

Multiview Differential Geometry in Application to Computer Vision

by

Ricardo Fabbri

B. S., Universidade de São Paulo, Brazil, 2003

M. S., Universidade de São Paulo, Brazil, 2004

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Division of Engineering at Brown University

Providence, Rhode Island

May 2011

© Copyright 2010 by Ricardo Fabbri

This dissertation by Ricardo Fabbri is accepted in its present form by the Department of Engineering as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____
Benjamin B. Kimia, Director
Division of Engineering

Recommended to the Graduate Council

Date _____
David B. Cooper, Reader
Division of Engineering

Date _____
Gabriel Taubin, Reader
Division of Engineering

Date _____
Peter J. Giblin, Reader
University of Liverpool

Approved by the Graduate Council

Date _____
Sheila Bonde
Dean of the Graduate School

Curriculum Vitae

Ricardo Fabbri was born on the 15th of July, 1980 in São José dos Campos, Brazil. During his middle school and high school studies he began experimenting with C++ programming inspired by his interests in video games. In 1999, he joined the ICMC institute from the University of São Paulo at São Carlos for his undergraduate degree in computer science. Since his freshman year, he has been an active computer vision researcher at the Cybernetic Vision Research Group from the Institute of Physics at São Carlos (IFSC), having published in prestigious conferences and at an international journal as an undergraduate student. He graduated in 2004 amongst the top three students in his class, and obtained a Masters degree in computer science also from ICMC in the same year. Since then he has been at the Laboratory for Engineering Man-Machine Systems (LEMS) at Brown University for his PhD studies, where he has worked with 3D scene reconstruction and camera calibration from multiple views using general curve structures. In 2008 he interned at Google, Inc., where he worked on rapid 3D scanning systems for books. He has published several papers in leading international journals. His research interests include computer vision, computational geometry, differential geometry, multiview stereo, camera auto-calibration, structure from motion, and related open source software.

Acknowledgements

First, and foremost, I dedicate this thesis to my father, Mauricio Fabbri, who taught me to find in science, and particularly in geometry, supreme joy and fun. Thank you, dad.

I would like to thank prof. Benjamin Kimia for the incredible enthusiasm in embracing my research topic. The most important lesson I learned from him is simply this: to value the clear expression of ideas. Never have I met someone more concerned with getting a clear message to the audience.

Thank you, Claudia Lopes, my mom, for being my teacher throughout my life. From you I learned how to read, I learned calculus, and, above all, to love.

This thesis would not have been completed without the amazing support and day-to-day companionship of my wife, Angela Guzmán. I consider her to have a honorary Ph.D. degree in computer engineering. Thanks, Angela, for having put up with my crazy schedule, absurd deadlines, endless anxieties, and other aspects of my wild graduate student life. Your generosity will always be remembered.

I would like to thank the rest of my family in Brazil, for having being so kind to put up with the long distance separation of one of their esteemed members. You can be sure I will be visiting you and keeping in touch with you more often from now on.

Thank you, prof. Luciano Costa, for being my academic mentor since undergraduate research in computer vision. I remember my first day working with you as a freshman, when my first assignment was to explore differential geometry of curves using Do Carmo's book. This thesis *is* an outgrowth of that study!

Thank you, prof. Odemir Bruno, for having encouraged me to produce a masters thesis, later published into a journal paper, prior to coming to the USA for my Ph.D. Your enthusiasm with science is also something that has inspired me a lot throughout the years.

Thank you, Daniela Sabino, for having been such an awesome friend and source of support in the USA. You have been a great source of inspiration since I saw your research

featured in national television prior to going to college.

Thank you, Leandro Estrozi, for being a great friend, and a true example of enthusiasm for pure science and its philosophy, and also for inspiring clear thinking into my research.

Thank you, Eduardo Almeida and Nhon Trihn, for being such warm friends and for allowing me to stay at your places while I was visiting from California.

Thank you, Anil Usumezbas, for being a great friend and research collaborator. I hope to keep in touch not only about our work, but also about classical music and movies.

I would like to thank all other graduate students at LEMS, for being so supportive and friendly. I had a great deal of fun with you guys.

Thanks to all professors at LEMS for the inspiring examples they set forth. I have learned a lot just from observing you, and I am proud to be graduating from such a competent lab composed of very approachable people. Thank you for all the advice you have given me.

Thank you Rich Minogue and Firat Kalaycilar for having scanned corrections to my papers and thesis while I was in California.

Thank you Jamie Kendrioski for the outstanding service you provided while at the Brown international office. Your kindness is something truly special.

Contents

List of Figures	xi
1 Introduction	1
1.1 Drawbacks of Isolated Point Features	3
1.2 Drawbacks of Intensity-Based Multiview Stereo	7
1.3 Advantages of Curves	9
1.3.1 Perceptual Motivation	10
1.4 Main Contributions	10
1.5 Organization of the Thesis	12
2 Previous Work	13
2.1 Curves for Calibration	13
2.1.1 Epipolar Tangencies	13
2.1.2 Infinitesimal Motion	20
2.2 Curves in Stereo	22
2.2.1 Multiview Stereo Using Curves	22
2.2.2 Binocular and Trinocular Stereo Using Curves	25
2.3 Occluding Contours for Surface Reconstruction in Video	27
2.4 Other Works Using Curves	40
3 Multiview Differential Geometry of Curves	45
3.1 Introduction	45
3.2 Notation and Formulation	51
3.2.1 Differential Geometry of Curves	51
3.2.2 Perspective Projection	51
3.2.3 Discrete and Continuous Sets of Views	53
3.2.4 Relating World and Camera-Centric Derivatives.	55

3.2.5	Stationary and Non-Stationary Contours	56
3.3	Projecting Differential Geometry Onto a Single View	56
3.3.1	Intrinsic Parameters and Differential Geometry	59
3.4	Reconstructing Differential Geometry from Multiple Views	60
3.5	Projecting Differential Geometry Under Differential Motion	63
3.5.1	Differential Relations for a Point	64
3.5.2	Differential Relations for a Curve	66
3.6	Projecting Differential Geometry of Surfaces	73
3.7	Conclusion	75
4	Camera Pose Estimation Using Curve Differential Geometry	77
4.1	Introduction	77
4.2	Determining Camera Pose from a Pair of 3D–2D Point-Tangent Correspondences	85
4.3	Calibrating Using Three Point-Tangent Correspondences Across Three Views	98
4.4	Future Directions	104
5	3D Curve Sketch: Curve-Based Stereo Reconstruction and Calibration	105
5.1	Introduction	105
5.2	Curve-based Multiview Stereo	110
5.3	Camera Calibration Refinement	118
5.4	Implementation	118
5.5	Experimental Results	121
5.6	Conclusion	123
6	Conclusion and Future Work	131
6.1	Future Work	132
6.1.1	Theory	132
6.1.2	Practice	132
A	Surface differential geometry	135
A.1	Basics	135
A.2	First Fundamental Form	136
A.2.1	Gradients on Surfaces.	138
A.3	Normals, Gauss Maps, and Second Fundamental Forms	138
A.3.1	Conjugacy	141

A.3.2	Gauss Map in Local Coordinates	142
A.3.3	Surface as the graph of a function $f(x, y)$	144
A.4	Useful identities for reference	146
B	Multiview Differential Geometry of Surfaces	147
B.1	Notation	147
B.2	Differential Relations: Rigid, Stationary Surface	151
B.3	Shape from Spatio-Temporal Shading	156
B.3.1	Considering Intrinsic Parameters	160
B.4	Derivation of the Shading Differential Equations	161
B.4.1	Derivatives of depth	165
B.5	Coefficients of model in camera frame	166
C	Bundle Adjustment	168
C.1	Introduction	168
C.2	Basic Formulation and Methods	168
D	Geometric and Qualitative Approach to Epipolar Geometry	173
D.1	Introduction	173
D.1.1	Motivation and Outline	173
D.1.2	Fundamental Concepts of Epipolar Geometry	174
D.1.3	The Essential Matrix	175
D.1.4	The Fundamental Matrix	177
D.1.5	Literature Overview	178
D.1.6	Classic Epipolar Geometry Estimation	178
D.2	Geometric Approach to Epipolar Geometry	180
D.2.1	Effect of Planar Transformations on Epipolar Geometry	180
D.2.2	Basic Geometric Aspects of Epipolar Homography	184
D.2.3	A Geometric Representation of Epipolar Geometry	188
D.2.4	Estimating the Epipolar Alignment Curve	189
D.2.5	Direct Approach to Solving the Geometric Equations	193
D.3	Qualitative Aspects and Epipolar Ordering	196
D.3.1	Partitioning via Ordering Invariance	196
D.3.2	Cost Function on Epipoles	200
D.3.3	Hierarchical Search	200
D.3.4	Contributions of this Section	200

D.3.5 Future work	202
Bibliography	203

List of Figures

1.1	Image curves arising from a range of physical sources.	2
1.2	A smooth scene for which the use of feature points breaks down. Row (a) shows two views of a Mazda concept car. In (b), interest points (SIFT features) were detected on each image, where the center of each circle denotes the position of the feature point. Notice how many regions of the car lack feature points. In (c) barely any features were properly matched, so that no camera estimation or reconstruction can be done. In contrast, the edge curves shown in (d) are much more dense, structural and representative of the car.	4
1.3	Another example of a smooth object for which the use of feature points breaks down. Column (a) shows a sculptural object in different views. Column (b) shows the detected feature points (SIFT). Notice how there are very few features detected in this object. In contrast, column (c) shows curves obtained from an edge detector and linker, which are capable of better representing the object.	5
1.4	Feature points are sparse: even for a textured object such as the picture on the left, there are regions with low texture which will not appear in the reconstruction.	5
1.5	The matching of point features such as SIFT breaks down with wide baseline. (a) shows the feature correspondence for small baseline. Notice how a considerable number of correct matches were found. However, when we increase the separation between the views (b), the matching breaks down. Other examples of failure are given in (c–d).	6
1.6	State-of-the-art methods based on interest points have been useful to determine camera models for each view, but result in a sparse and unorganized point cloud reconstruction (right). These results are from [1].	6

1.7	Example of the failure of interest point matching near high surface curvature with foreshortening. The top row shows two views of an indigenous bird sculpture, which is highly textured and curved. The bottom row shows feature points detected in each view, and the feature points that were properly matched between the two views are shown in green. Matching did not occur near regions curving away from the viewer.	8
1.8	From [74]: the typical output of multiview stereo methods give detailed textured 3D reconstruction of objects (right), but operate under controlled conditions (<i>e.g.</i> , homogeneous background), and are often initialized by the visual hull of the object (left), which requires well-segmentable silhouettes in each view and presumes a well-defined working volume.	9
1.9	From [29]: subjects are able to judge 3D surface normals of an object depicted by a line-drawing almost as accurately as for objects depicted by a shaded image, indicating that image curves depict much of the image information.	10
1.10	From [17]: Wheatstone-influenced curved line drawing stereogram.	11
2.1	Correspondence of epipolar tangencies used in curve-based camera calibration. An epipolar line on the left, whose tangency at a curve is marked in a certain color, must correspond to the epipolar line on the right having tangency on the corresponding curve, marked with the same color. This concept works for both static curves and occluding contours.	14
2.2	Illustrating the differential update of epipolar tangencies through the use of the osculating circle or curvature information.	15
2.3	From [140]: given hypothesized epipoles, three corresponding epipolar lines are needed to solve for the complete epipolar geometry. The authors easily obtain two such correspondences at an instant $t = t_0$ (silhouette in gray) though the outermost epipolar tangencies, and a third correspondence is obtained by keeping the camera fixed but taking a snapshot at $t = t_1$ and using an outer epipolar tangency (at the white silhouette).	20
2.4	Figure illustrating how the stereo and motion constraints are used in [22].	42

3.1	(a) Interest point-based approaches give a sparse reconstruction in the form of an unorganized cloud of points; image taken from [1]. (b) Views with wide baseline separation may not have any interest points in common, but they often do share common curve structure. There may not always be sufficient interest points matching across views of homogenous objects such as the car in (c) or the sculpture in (d), but there is sufficient curve structure. (e) each moving object requires its own set of features, but they may not be sufficient without a rich texture surface. (f) Non-rigid structures face the same issue.	47
3.2	(a) The perspective projection of a space curve. (b) The projection of a space curve in n views.	52
3.3	Multiview formulation of continuous camera motion and a possibly moving contour.	54
3.4	3D Tangent reconstruction from two views as the intersection of two planes.	62
4.1	(a) Views with wide baseline separation may not have any interest points in common, but they often do share common curve structure. There may not always be sufficient interest points matching across views of homogenous objects such as for the sculpture in (c), but there is sufficient curve structure. (b) Each moving object requires its own set of features, but they may not be sufficient without a rich texture surface. (d) Non-rigid structures face the same issue.	79
4.2	Challenges in using curve fragments in multiview geometry: (a) instabilities with slight changes in viewpoint, as shown for two views in (b) and zoomed in selectively in (c-h) real examples of edge grouping instabilities, such as a curve in one being broken into two in another view, a curve being linked onto background, a curve being detected in one view but absent in another, a curve being fragmented into various pieces at junctions in one view but fully linked in another view, different parts of a curve being occluded in different views, and a curve undergoing shape deformation from one view to the other. (i) Point correspondence ambiguity along the curve.	80
4.3	The problem of finding the camera pose \mathcal{R} , \mathcal{T} given space curves in a world coordinate system and their projections in an image coordinate system (a), and an approach to that consisting of (b) finding the camera pose \mathcal{R} , \mathcal{T} given 3D point-tangents (<i>i.e.</i> , local curve models) in a world coordinate system and their projections in an image coordinate system.	81

4.4	Interpretation of the essential matrix equation. All vectors are written with respect to the 3D coordinate system of camera 2.	82
4.5	Correspondence of epipolar tangencies used in curve-based camera calibration. An epipolar line on the left, whose tangency at a curve is marked in a certain color, must correspond to the epipolar line on the right having tangency on the corresponding curve, marked with the same color. This concept works for both static curves and occluding contours.	83
4.6	Illustrating the differential update of epipolar tangencies through the use of the osculating circle or curvature information.	83
4.7	From [140]: given hypothesized epipoles, three corresponding epipolar lines are needed to solve for the complete epipolar geometry. The authors easily obtain two such correspondences at an instant $t = t_0$ (silhouette in gray) though the outermost epipolar tangencies, and a third correspondence is obtained by keeping the camera fixed but taking a snapshot at $t = t_1$ and using an outer epipolar tangency (at the white silhouette).	84
4.8	Diagram of the mutual intersection of Equations 4.2.24–4.2.26 in the x_1 – x_2 plane.	92
4.9	Diagram illustrating a parametrization of the ellipse by a parameter t	96
4.10	Variants of the multiview camera geometry estimation problem.	99
4.11	Intrinsic coordinate system is in brown, while the view coordinate system is in green.	101
4.12	102
5.1	The 3D reconstruction as a 3D curve sketch (left) gives explicit geometry in contrast to the 3D cloud of point reconstruction (right). In essence, the proposed method produces a cloud of 3D curve fragments.	106
5.2	Challenges in using curve fragments in multiview geometry: (a) point correspondence ambiguity along the curve, (b) instabilities with slight changes in viewpoint, (c-h) real examples of edge grouping instabilities, such as a curve in one being broken into two in another view, a curve being linked onto background, a curve being detected in one view but absent in another, a curve being fragmented into various pieces at junctions in one view but fully linked in another view, different parts of a curve being occluded in different views, and a curve undergoing shape deformation from one view to the other.	107

5.3	A schematic overview of our approach: The input to our system is (i) a set of N images taken of the scene where N is at least 6 but typically larger, and (ii) a set of N coarsely calibrated camera models, one for each view: the system pairs a curve fragment from an “anchor view” with one from another view, “hypothesis views”, to form a curve pair correspondence hypothesis which is verified or discarded based on the edge map of other views, “confirmation views”. Once a set of reliable curve fragment hypotheses are formed, curve-based bundle adjustment is used to refine the cameras and the process is repeated. The output of our system is a set of 3D curve fragments (the 3D curve sketch) and refined cameras.	109
5.4	A putatively corresponding set of image curve fragments can arise from a single 3D curve.	109
5.5	(a) Each space curve, or at least a significant subsegment of it, is expected to have corresponding image curve subsegments in at least two views. (b) An “epipolar band” for a curve segment is the set of epipolar lines going through points of that curve segment. The epipolar band of image curve $\gamma_{i_1}^{v_1}$ (green and yellow) delineates potential image curve matches by requiring some overlap in this region, allowing for $\gamma_{i_1}^{v_2}$ and $\gamma_{i_3}^{v_2}$ but not $\gamma_{i_4}^{v_2}$. Conversely, the epipolar band of $\gamma_{i_2}^{v_2}$ (green and blue) selects $\gamma_{i_1}^{v_1}$, and others. The correspondence between $\gamma_{i_1}^{v_1}$ and $\gamma_{i_2}^{v_2}$ is restricted to the subsegments a_1b_1 and a_2b_2 , respectively, <i>i.e.</i> , the common (green) epipolar band, required to exceed a minimal epipolar overlap ($\tau_e = 5$ pixels).	111
5.6	Real examples of how instabilities in the edge linking process make it difficult to evaluate the support of a reprojected curve (light blue) using curve fragments (random colors).	112
5.7	A single edge hypothesizes a vast space of curves, while a pair of edges hypothesize a reduced space, a triplet of edges represent a more restricted set of curves, <i>etc.</i>	113
5.8	An example of curvelets detected in a real image (a side window of the Capitol building depicted in Figure 5.2): (a) The original edges; (b) all the curvelets representing local groupings in a 5×5 neighborhood, and (c) all the curvelets representing local groupings in a 7×7 neighborhood.	113
5.9	Among the discrete curvelets (red and light green) at an edge point (yellow with magenta dot), only one (green) matches the reprojected curve $\gamma^{u,v}$ (light blue). The remaining edges in the edge map are shown in darker blue.	114

5.10	We avoid multiple intersections with epipolar lines by breaking curve fragments at epipolar tangencies.	114
5.11	The instability of the edge linking process can be altogether avoided if we use edges, as done in recent object recognition strategies, <i>e.g.</i> , oriented chamfer distance [119, 139, 149]. However, reprojected contours (blue) can at times get significant support from cluttered or textured areas (a). This motivates the exclusive use of edges for which there exists a local grouping consistent with the reprojected curve (b). Two of the edges with a magenta dot in (b) support the reprojected curve with a curve model (green) while two others do not (red).	115
5.12	The reprojection of a curve fragment pair $\gamma_{i_1}^{v_1}, \gamma_{i_2}^{v_2}$ onto a third view (shown in blue) can enjoy the support of an existing curve fragment γ_i^v (red) as in (a). However, much more typically, due to the instability of the edge linking process, it is more likely to expect support from several pieces of curve fragments as in (b), with missing segments. This instability motivates the use of a small neighborhood of an edge – not entire curve fragments – in the confirmation process.	117
5.13	Example of a projected curve $\gamma^{k,v}$ (red) on edge map M_v (blue) before calibration refinement (a) and after (b).	119
5.14	(a) A set of sample views from the “Capitol Sequence”; (b) detected subpixel edgels at low gradient threshold are used to generate hypothesis curves; (c) detected subpixel edgels at a higher gradient threshold are used in the confirmation views; (d) curve fragments with length > 40 pixels used in Stage I; (e) fragments with length > 20 pixels; (f) a selected curve is highlighted in view 1 and the curves in view 2 with sufficient epipolar overlap are marked in blue. In the next three images the reprojected hypothesis curve pair is shown in cyan, confirmed in (g) but rejected in (h) and (i). The example in (i) shows the need for orientation in the presence of clutter!	120
5.15	Example of human-marked ground truth correspondences on a small section of the “Capitol Sequence”. Curves that were marked as corresponding are represented with the same color, and curves not participating in any correspondence are in red. The thicker blue curve in the left image indicates a curve the user is inspecting.	122

5.16	Precision/Recall curves for the core 3D curve sketch correspondence, obtained by varying the total support score τ_t . Left: the optimal evaluation curve shown in red; Right: varying the orientation threshold τ_θ to demonstrate the usefulness of the differential geometry constraint. Notice how eliminating orientation ($\tau_\theta = 90^\circ$) degrades performance. These experiments used disambiguation threshold $\tau_r = 1.5$, length $\tau_l = 40px$ for the Downtown and Capitol sequences (hi-res), and $\tau_l = 20px$ for the Middlebury Dino sequence (low-res).	124
5.17	The 3D curve sketches for the Dino and Capitol sequences.	125
5.18	The 3D curve sketch reconstructions for the Downtown sequence using 30 sets of hypothesis views: (a) the raw 3D curve sketch showing the best candidate curves without any support confidence threshold ($\tau_t = 0$); (b-c) top and side views of the reliable curve sketch under the operating point of 30% precision and 100% recall ($\tau_t = 100$ supporting edgels). The dots seen in (b) are top views of perfectly vertical curves of buildings. The representation is very efficient while still being rich enough to be used in applications such as calibration refinement and for registering new views to the scene.	126
5.19	Raw 3D curve sketch reconstruction for the dinosaur with maximum recall, showing the raw output of the best candidate curves before thresholding for support. Figure 5.17, in contrast, shows only the curves at the 30% recall rate operating point.	127
5.20	Additional 3D curve sketches for the capitol: (a) a view of the reconstruction for the maximum recall, showing all the best curves along the epipolar lines, without any threshold ($\tau_t = 0$) on the total support; (b) the same reconstruction operating at 30% recall rate and perfect precision (see Figure 8 in the original paper for the operating point parameters), being more sparse but much more reliable, used as input to curve-based calibration refinement. (d) The same reconstruction at 50% recall rate, which yields 100% precision in the evaluation, but which might contain false positives here because we run the system for many 30 hypothesis views, not including the ones used in evaluation.	128
A.1	Orthogonal (thus conjugate) directions r_1 and r_2 transform to conjugate directions r'_1 and r'_2 after an affine or projective transformation.	142

B.1	Projection of a quadric in images.	148
B.2	(a) Spherical coordinates used in this text, and (b) The orientation of a (tangent) plane with respect to the camera plane can be given by the <i>slant</i> – which is the angle between the normal of the plane and the viewing direction – and the <i>tilt</i> – which is the angle the projected normal makes with the <i>x</i> -axis of the camera coordinate system (from [55]).	149
C.1	Form of the Jacobian matrix for bundle adjustment estimating cameras and 3D points.	171
D.1	Interpretation of the essential matrix equation. All vectors are written with respect to the 3D coordinate system of camera 2.	177
D.2	Skewed image coordinates	182
D.3	Spherical coordinates of \mathbf{N}_2 in the universal frame.	186
D.4	Ordering invariant.	197
D.5	Polygon partitioning of the space of putative epipoles: (a) restricted to the two image domains, and (b) restricted to a more extended region of the 2D plane with the true epipole shown as a small square. Polygons are color-coded to represent distinct equivalence classes of putative epipoles. Black regions represent impossible solutions violating ordering. Notice how the true epipole satisfies the ordering represented by the blue polygon.	198
D.6	Stability plots. Even a simple solution of epipolar geometry using just the centroids of our polygons clearly outperforms the 8 point algorithm for medium to large localization error levels.	201

Chapter 1

Introduction

Given a sequence of 2D images, such as a video or a set of unorganized pictures, what can be inferred about the 3D scene? Where was the camera at each picture? What were the camera settings such as zoom level or focal length? These are fundamental problems in 3D Computer Vision and Photogrammetry, and have found a wide range of applications such as match-moving in cinematography (*e.g.*, the effects in *The Matrix*), the organization of a collection of photographs with respect to a scene (*e.g.*, Phototourism [1] and the Look Around feature in Google Panoramio), robotic manipulation, metrology from cameras in the automobile industry, and 3D modeling from photographs for architecture, electronic gaming, archaeology, and urban modeling (*e.g.*, Google Streetview).

In this thesis, we develop the multiple view geometry of arbitrary, piecewise differentiable curves, and the beginnings of the local theory on general surfaces. Curves occur everywhere around us as outlines of objects, markings on surfaces, ridges and valleys, shadow boundaries, and thin objects such as branches and wires. Figure 1.1 shows different types of image curves arising as contrast edges, which are primary types of curves we will be concerned with. Other curves that benefit from the theory in this thesis are trajectories of moving particles or markers observed by a system of cameras.

Curves are important on their own, as argued later in this chapter, but, most importantly, they are key as the initial part of a broader effort to reconstruct and parse general, complex 3D scenes. We envision a large system where curves are used to bootstrap camera pose for each image of a given sequence, and to generate an initial curve-based reconstruction of the scene which we call a *3D curve sketch*. The reconstructed curves form the foundations on top of which detailed 3D surfaces may be constructed for a general scene. In this thesis we concentrate on curves, although we give preliminary results on local surfaces

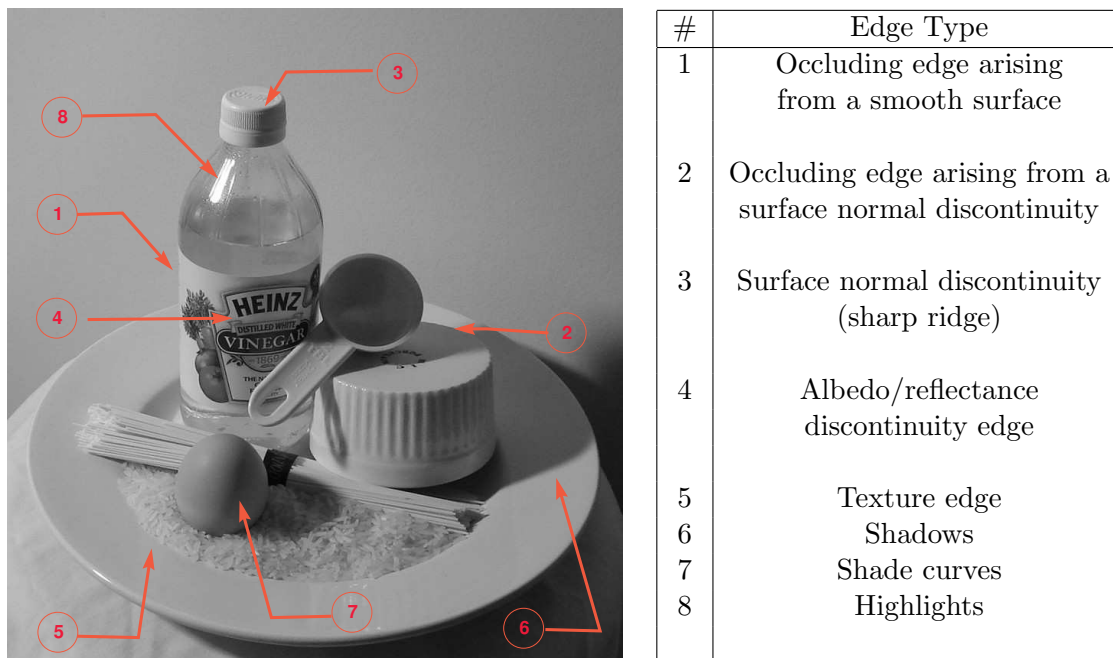


Figure 1.1: Image curves arising from a range of physical sources.

and shading in multiple views.

In previous work, there has been very little treatment of general curves. The state of the art in structure from motion, which solves for camera models in each view and generates a preliminary 3D reconstruction, is based on sparse point features and, to a much lesser extent, on straight lines and other algebraic curves such as conics. On the other extreme, state-of-the-art multiview stereo methods, which reconstruct detailed 3D surfaces of simple objects given precise camera models, operate on a pixel intensity level. These approaches have been successful for select domains of application, resulting in autocalibration and useful 3D reconstructions. However, as described later in this chapter, these methods and their representations have drawbacks. General curves provide an intermediate, rich, and meaningful representation being more succinct and structured than pixels or voxels, while more descriptive and representative of scene content than isolated feature points.

The study of general piecewise smooth curves and surfaces represents a branch of mathematics known as *differential geometry*. Our approach advocates the use of differential geometry to model general curves and surfaces instead of plain projective geometry, which is well-suited to model only points and globally algebraic curves such as lines and conics, and algebraic surfaces such as planes and quadrics. In a sense, our approach proposes to

augment multiple view geometry with differential methods in much the same way that analytical geometry is traditionally augmented with calculus in order to model general curved phenomena through local analysis. Differential geometry allows to create local models for integrating local information with information in a neighborhood, the basis of geometric consistency. Our initial inspiration for this approach was partly drawn from the work of Li and Zucker on curve-based binocular stereo [94].

We now analyze the drawbacks of state-of-the-art approaches, and the advantages of curves.

1.1 Drawbacks of Isolated Point Features

The application of **interest-point-based methods** have been successful in scenes with texture-rich images [1, 125]. Despite their success, these methods are not applicable in general settings for reasons detailed below.

- They assume an abundance of interest points per independently moving object, possibly only with a fair degree of texture in projected images. Such a rich texture is not always available in scenes with homogeneous regions, as in some man-made environments, as illustrated in Figures 1.2 and 1.3, or when objects project to a small number of pixels, relative to the basis of computation of interest points. Even for objects that are overall well-textured, having enough point features to recover low-parametric motion parameters, there are often parts which lack texture and cannot be reconstructed, as shown in Figure 1.4. In all these cases, there may be sufficient image curve geometry, however, to reconstruct the scene and recover the cameras. Moreover, when modeling non-rigid objects, point features are not dense enough, since there is not enough of them per local patch.
- The stability of interest points is significantly reduced as the baseline exceeds 30° [113], so these methods are fundamentally restricted to a limited range of baselines. Figure 1.5 illustrates this point. In contrast, certain curve features, such as sharp ridges on a building, persist over a much greater range of views.
- Reconstruction using interest points results in an unorganized cloud of 3D points where the geometric structure of the underlying curves and surfaces is not explicit, as shown in Figure 1.6. This is not much of a problem if the focus is on calibrating the viewpoints as in Phototourism [1]. However, when the 3D object geometry is

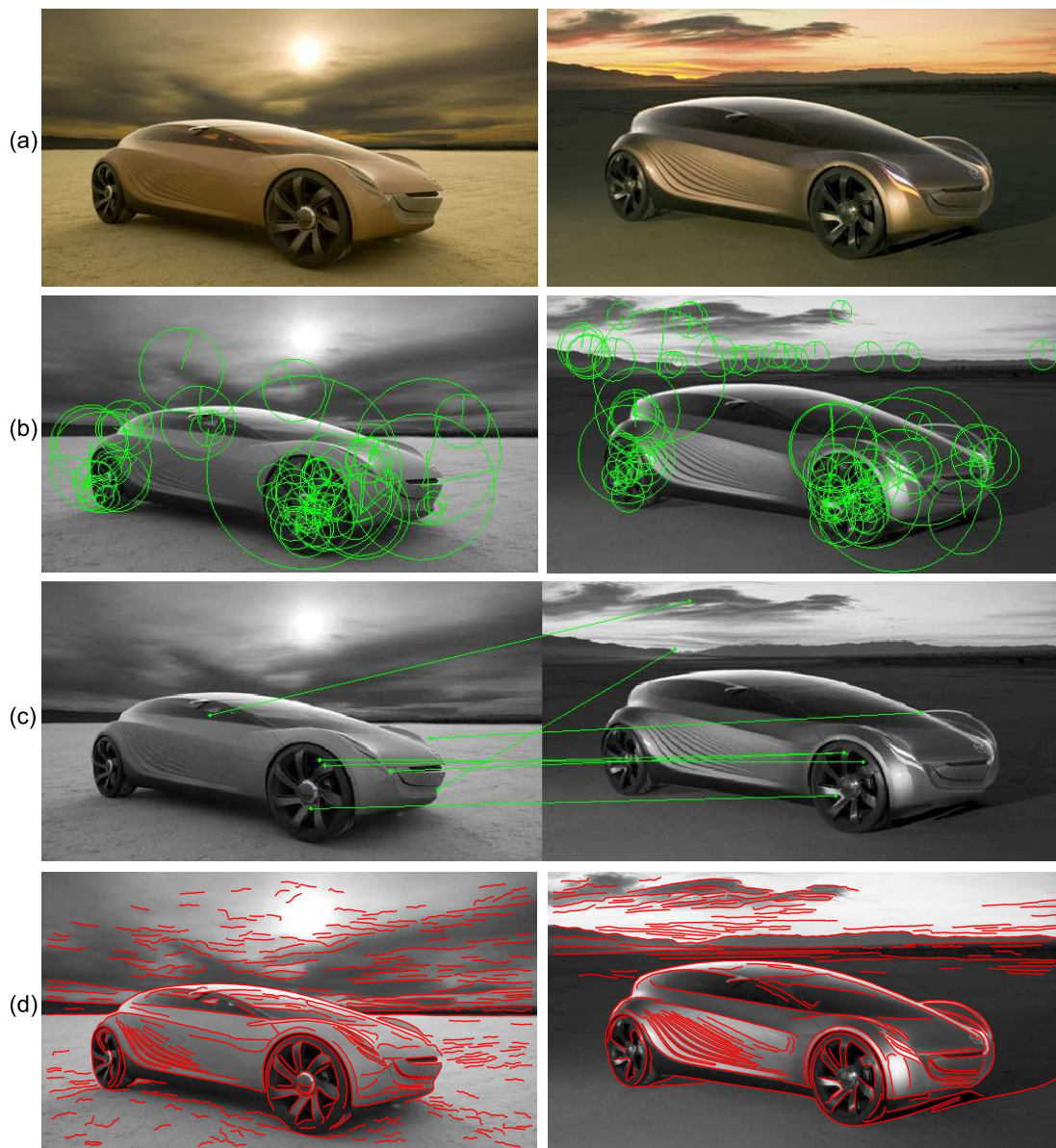


Figure 1.2: A smooth scene for which the use of feature points breaks down. Row (a) shows two views of a Mazda concept car. In (b), interest points (SIFT features) were detected on each image, where the center of each circle denotes the position of the feature point. Notice how many regions of the car lack feature points. In (c) barely any features were properly matched, so that no camera estimation or reconstruction can be done. In contrast, the edge curves shown in (d) are much more dense, structural and representative of the car.

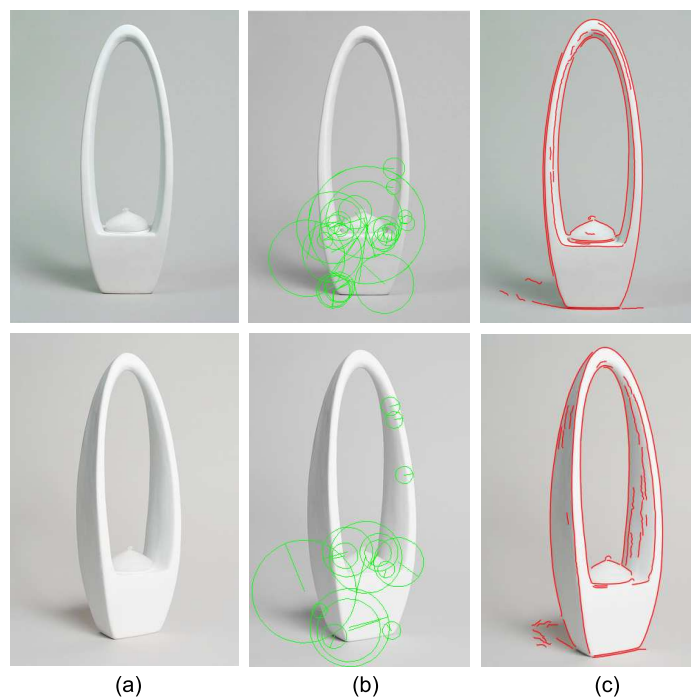


Figure 1.3: Another example of a smooth object for which the use of feature points breaks down. Column (a) shows a sculptural object in different views. Column (b) shows the detected feature points (SIFT). Notice how there are very few features detected in this object. In contrast, column (c) shows curves obtained from an edge detector and linker, which are capable of better representing the object.

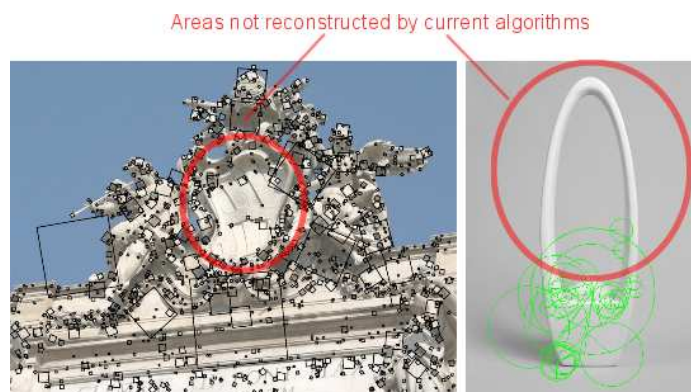


Figure 1.4: Feature points are sparse: even for a textured object such as the picture on the left, there are regions with low texture which will not appear in the reconstruction.

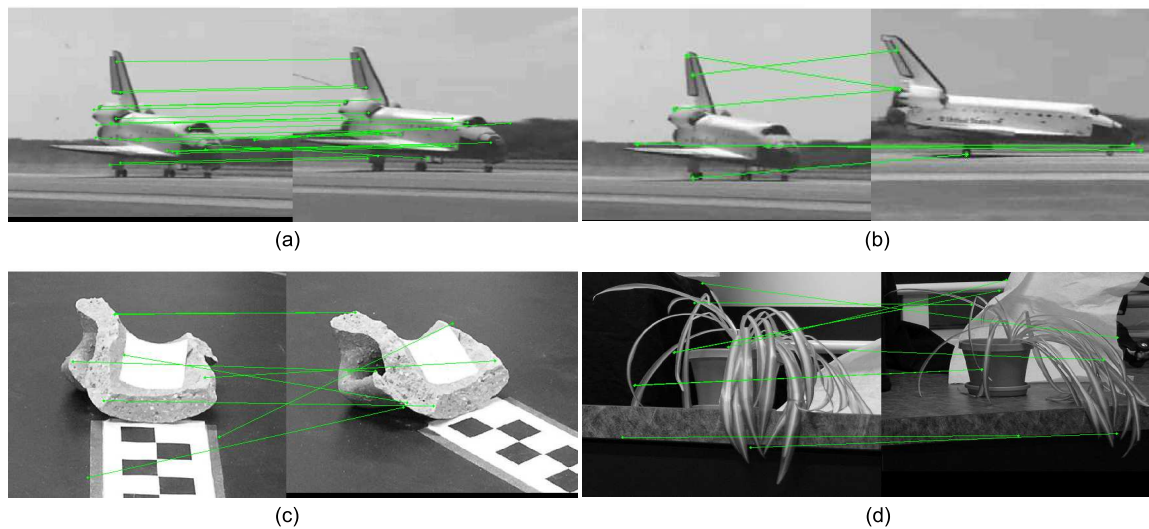


Figure 1.5: The matching of point features such as SIFT breaks down with wide baseline. (a) shows the feature correspondence for small baseline. Notice how a considerable number of correct matches were found. However, when we increase the separation between the views (b), the matching breaks down. Other examples of failure are given in (c–d).

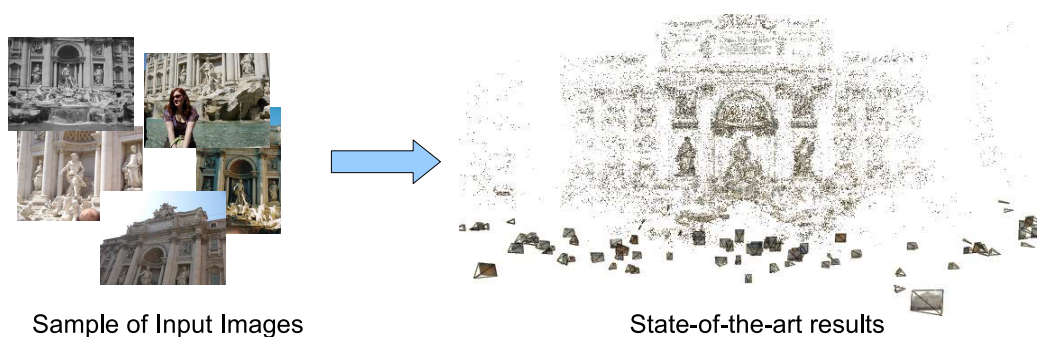


Figure 1.6: State-of-the-art methods based on interest points have been useful to determine camera models for each view, but result in a sparse and unorganized point cloud reconstruction (right). These results are from [1].

required, such as in registration of new views, modeling for architecture, archaeology, entertainment, object recognition, and robotic manipulation, it would be useful to augment the output with an explicit geometric structure of curves. It is worth noting that it is common to mesh a 3D point cloud, particularly using Poisson surface reconstruction [87] when normals are computed at each 3D point using patch-based multiview stereo techniques [58]. This approach suffers from oversmoothing artifacts, specially in regions where the 3D points are sparse.

- Matching of interest points breaks down at regions with considerable surface curvature and foreshortening, such as parts of surfaces curving away from the viewer. This is due to the fact that the change from one view to the other can no longer be modeled by a simple transformation as assumed by point features (in the case of SIFT, this is a rotation and scaling), but the transformation needs to account for surface curvature. Therefore, these regions cannot be reconstructed, as shown in Figure 1.7.

1.2 Drawbacks of Intensity-Based Multiview Stereo

Another category of 3D reconstruction techniques, **multi-view stereo methods**, produce detailed 3D reconstructions of objects imaged under controlled conditions by a large number of precisely calibrated cameras [56, 61, 64, 74] (see [134] for a review). However, they cannot handle general scenes, since most approaches are restricted to a single object or to a specific type of object, such as buildings. In addition, they often require accurate camera calibration, operate under controlled acquisition, and are often initialized by the visual hull of the object or a bounded 3D voxel volume. Figure 1.8 shows an example.

A large body of literature in **two-view short-baseline stereo** are based on correlating unorganized and sparse feature points by matching some aspects of the local region surrounding the feature to disambiguate correspondences, and other methods find a dense depth map by correlating a local neighborhood of each pixel. A number of criteria such as smoothness, uniqueness, ordering, limited disparity, and limited orientation disparity, have been used to deal with the inherent ambiguity. However, these can break down, especially with wide baseline, with multiple nearby structures, or when discontinuities and branching structures exist [94]. Moreover, most dense binocular stereo approaches assume the surfaces are locally fronto-parallel, which can generate spurious reconstructions for curved surfaces. This was remedied by Li and Zucker [96] through the use of surface differential geometry, as advocated in this thesis in a more general setting.

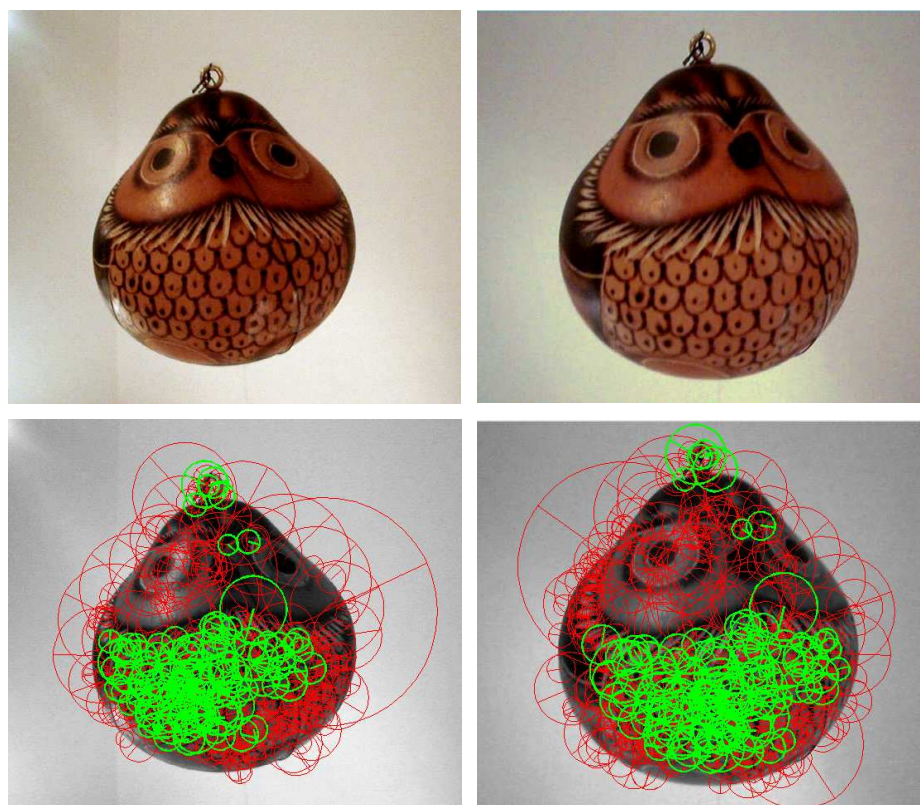


Figure 1.7: Example of the failure of interest point matching near high surface curvature with foreshortening. The top row shows two views of an indigenous bird sculpture, which is highly textured and curved. The bottom row shows feature points detected in each view, and the feature points that were properly matched between the two views are shown in green. Matching did not occur near regions curving away from the viewer.



Figure 1.8: From [74]: the typical output of multiview stereo methods give detailed textured 3D reconstruction of objects (right), but operate under controlled conditions (*e.g.*, homogeneous background), and are often initialized by the visual hull of the object (left), which requires well-segmentable silhouettes in each view and presumes a well-defined working volume.

1.3 Advantages of Curves

“By reporting only the lines of contrast, the retina avoids having to prepare endless, uninteresting and massively redundant reports about plain surfaces”– Simon Ings in A Natural History of Seeing

The goal of this thesis is to augment current multiview reconstruction and camera calibration technology by developing a generally applicable approach based on curves, which is attractive because:

- Curves arising from edge discontinuities are denser and more structured than interest points, while efficiently representing the image or 3D scene. They provide a useful middle ground between a costly and redundant pixel/voxel array representation and a very sparse, unstructured point cloud representation. This allows the fast production of reconstructions that, although not meshed, are still recognizable, structured and cheap to manipulate and store. For instance, it is impossible to register new cameras given just a 3D point cloud, but given a 3D curve representation of the scene this task becomes possible due to the added structure. It is well-known that edge-based representations can efficiently represent most of the image content [39], and this motivates an efficient 3D curve-based reconstruction.
- Curves have greater invariance than interest points to changes in illumination.
- Curves are stable over a greater range of baselines as compared to interest points.
- Curves have good localization in the orthogonal direction at each point, and their long extent allows for more accurate detection and localization under a wider variety of viewpoint changes than point features.

- Edge curve structure is correlated with surface properties. The reflectance or ridge curves provide boundary condition for surface reconstruction, while occluding contour variations across views indicate surface properties [27].
- Long and complex enough curves are less ambiguous to match and track since they are richer and more structured primitives than points and lines.

1.3.1 Perceptual Motivation

The notion that image curves contain much of the image information can be supported by a recent study [29] which found that generally subjects are able to judge 3D surface normals of an object depicted by a line-drawing almost as accurately as for objects depicted by a shaded image, Figure 1.9. Furthermore, some indication of the importance of curves in human stereo vision comes from the

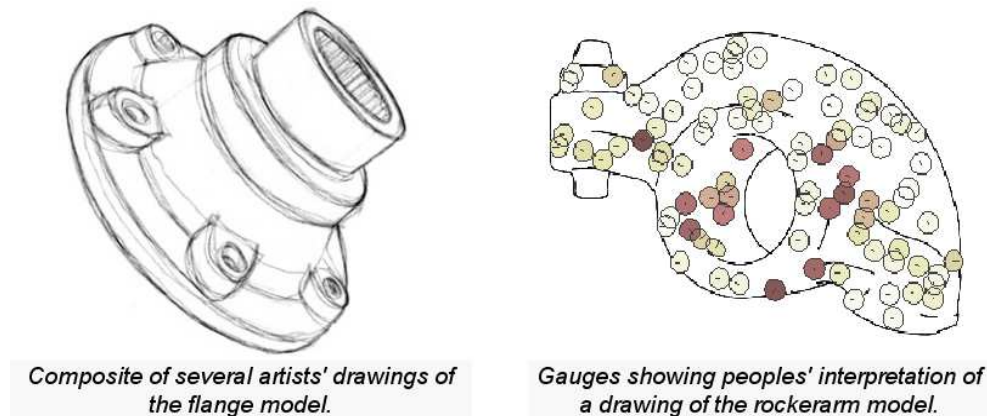


Figure 1.9: From [29]: subjects are able to judge 3D surface normals of an object depicted by a line-drawing almost as accurately as for objects depicted by a shaded image, indicating that image curves depict much of the image information.

line drawing stereograms of Wheatstone and his successors [156], where it seems that the *shapes* of the curves are crucial in determining the stereo correspondences, Figure 1.10.

1.4 Main Contributions

The contributions of this thesis are as follows.

On the theoretical side (Chapters 3 and 4), our main contribution is the development of a complete study of the local geometry of curves in multiple views, as well as the beginnings of a similar study for surfaces. We provided a unified framework for modeling curvilinear structure in multiple views, covering fixed, occluding, and nonrigid curves, for the cases of a discrete set of views and differentiable camera motion. This framework unifies and corrects previously scattered results in the literature, and also provides new results. We also developed a preliminary study on the case

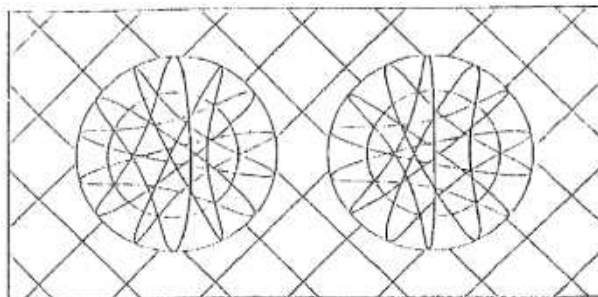


Figure 1.10: From [17]: Wheatstone-influenced curved line drawing stereogram.

of curved surfaces, their local shading and the local behavior of surface markings. The theory is of fundamental importance to any application requiring relationships between images and 3D structure of curves, such as stereo matching, curve tracking, and camera estimation.

We devised formulas for obtaining the differential-geometric quantities of tangent, curvature, and curvature derivative of an image curve in terms of the tangent, curvature, curvature derivative, and torsion of the underlying space curve (Chapter 3). We also showed how tangent, curvature, curvature derivative, and torsion of a fixed space curve can be reconstructed from two views. Torsion is essential in the modeling of non-planar curves, and to the best of our knowledge, this is the first time torsion reconstruction from multiple views appears in the literature. Moreover, we showed how to relate the parametrization of a projected curve to the parametrization of the space curve, and the parametrizations of two perspective projections of the same space curve. Our approach provides new derivations for previous lower-order results [50,94] using a simpler method that easily generalizes to the higher order results involving curvature derivative and torsion.

Another contribution concerns the unified study of how contour image velocities and their spatio-temporal derivatives behave in terms of differentiable camera motion parameters and 3D motion parameters, for the cases of rigid/stationary curves, occluding contours, and general non-rigid curves (Chapter 3). Among the most important results, we provide an extension and correction of a result from [120,122] regarding a polynomial equation that relates observed normal contour velocities to differentiable camera motion. The original equation was corrected for a missing term, and we extended it to handle occluding contours.

We also devised a method for extrinsic camera calibration (also known as camera pose estimation or camera resectioning) from edgel correspondences. In other words, given enough 3D-2D correspondences between points having attributed tangents, the method solves for the camera pose. This is an important step toward camera calibration from the local geometry of curves, and the method is intended for use as a minimal engine within a RANSAC framework.

On the practical side (Chapter 5), we developed an approach, called the 3D curve sketch, to match and reconstruct curves from multiple views of general scenes. The method is able to efficiently integrate information across a very large number of views. It requires initial camera models which are usually imprecise, and refines the cameras using curve fragments and edge information. The

approach augments existing interest point based and stereo approaches in providing explicit curve geometry as well as in extending applications where the assumption of these methods fail but image curve content is significant. The work is expected to form the initial building block in a broader effort to use image evidence of the explicit geometry of curves and surfaces and reconstruct these by integrating information across many views in a flexible way. The 3D curve sketch will be the initial structure on which surfaces may be hypothesized and constructed.

1.5 Organization of the Thesis

In Chapter 3.2 we lay down the basic notation and definitions regarding the classic differential geometry of curves, how we model perspective projection, and how we model the cases of a discrete set of views observing a scene and the kinematics of continuous camera motion (as in video). In Chapter 2 we review the literature on the use of curves in multiple view geometry, rewriting some important results using our notation.

Chapter 3 covers the main results of this thesis, namely the study of projection and reconstruction of curve differential geometry. In Section 3.3 we study how differential geometry of a space curve relates to the differential geometry of its perspective projection. The way these differential-geometric quantities are affected by the intrinsic parameter transformation is studied in Section 3.3.1. In Section 3.4 we show how the differential-geometric quantities of tangent, curvature, curvature derivative, and torsion of a fixed space curve can be reconstructed from two views. We also show how to relate the parametrization of projected curves in two views to each other.

Chapter 3 also studies the differential geometry of curves under differentiable motion. It covers how the image velocities and their spatio-temporal derivatives behave in terms of camera motion parameters and 3D motion parameters, for the cases of rigid/stationary curves, occluding contours, and general non-rigid curves. Appendix B covers preliminary material on differential-geometric relationships for surfaces, features lying on surfaces, and shading. The purpose of this appendix is mainly for future work. In Chapter 4 we give the theory for extrinsic camera calibration (camera resectioning) given 3D-2D correspondences between points with attributed tangents. In Chapter 5 we describe a practical method for curve-based reconstruction and camera calibration, called the 3D curve sketch.

Chapter 2

Previous Work

This chapter reviews previous work on the use of curves in multiple view geometry. The literature can be divided into curve-based camera calibration and curve-based stereo reconstruction. The calibration category can be subdivided into methods that make use of epipolar tangencies, and methods that calibrate from tracked curves in video. The stereo category can be subdivided into multiview stereo (which uses many views), binocular and trinocular stereo, and occluding contours for surface reconstruction in video. Papers that don't fit neatly into the above categorization (*e.g.*, papers that cover many aspects of them) are put in an 'other works using curves' category. We now delve into each of these in detail. For each sub-category of approaches, we provide an overview of the literature and the underlying concepts, summarize key selected papers in further detail, and list the main drawbacks. For details on the notation used in this chapter, see Chapter 3.

2.1 Curves for Calibration

2.1.1 Epipolar Tangencies

The problem attacked by these papers is: given corresponding image curves across two or more views, but not their alignment (*i.e.*, point-wise correspondence), obtain the multiview epipolar geometry (or camera pose if the intrinsic parameters are known).

Very little work has been done to tackle this problem, which boils down to finding the epipolar geometry consistent with the given curve correspondences. All previous work in this category are restricted to closed curves, which can be either images of fixed curves or of occluding contours. It can be shown that in this case the constraints on calibration occur at so-called epipolar tangencies: if an epipolar line is tangent to a curve in one image, its corresponding epipolar line must also be tangent to the same curve in the other image, Figure 2.1.

Porrill and Pollard [126] started the concepts of how epipolar tangencies constrain epipolar geometry, and devised an iterative method to refine the epipolar geometry once the solution is sufficiently

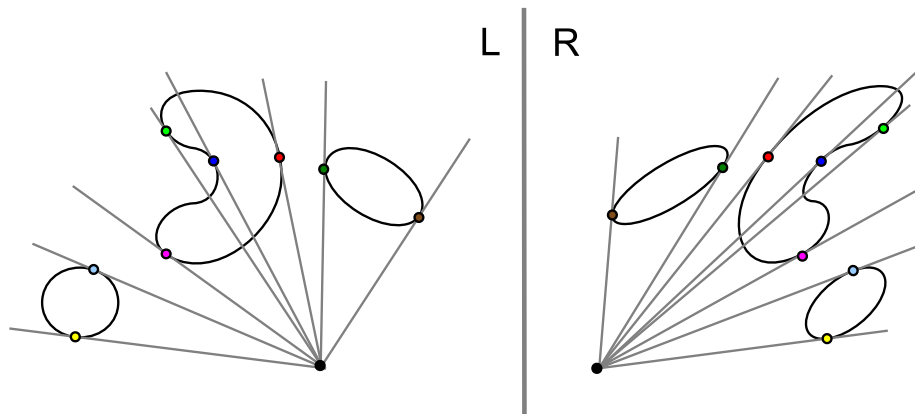


Figure 2.1: Correspondence of epipolar tangencies used in curve-based camera calibration. An epipolar line on the left, whose tangency at a curve is marked in a certain color, must correspond to the epipolar line on the right having tangency on the corresponding curve, marked with the same color. This concept works for both static curves and occluding contours.

close to the true one. They make use of curvature in order to obtain a local osculating circle approximation to the curve at epipolar tangencies in order to efficiently update the tangency position after each iteration. This method is described in more detail in Section 2.1.1. Kahl and Heyden [85] attempted to find a closed-form solution to epipolar geometry given curve correspondences, and they succeeded for the case where the curves are conics. Kaminski and Shashua [86] provided closed-form solutions to the case of general algebraic curves. Berthilsson *et. al.* [13] provided an iterative method and a non-optimal way of finding an initial solution, but no closed form is given.

The same ideas are also valid in the case of occluding contours such as object silhouettes [5, 6, 26, 59, 73, 109, 130, 140, 157, 158]. In this context, epipolar tangencies are also known as frontier points. Silhouettes are attractive in controlled conditions with a large number of cameras observing an object (*e.g.*, a human figure) where the background can be suitably chosen to make silhouette extraction easy. In these situations the silhouettes are visible from any direction and their epipolar tangencies are useful for calibrating the views.

The main drawbacks of previous work on curve-based camera calibration using epipolar tangencies are:

- All methods assume closed curves
- Global algebraic curve models are unrealistic for general scenes
- Iterative methods require initialization
- Silhouette methods require controlled acquisition.

We now review selected papers in further detail.

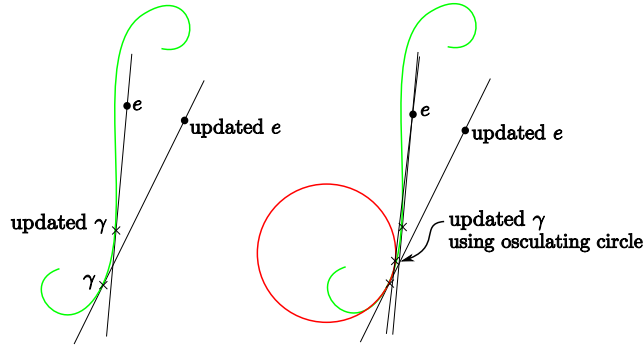


Figure 2.2: Illustrating the differential update of epipolar tangencies through the use of the osculating circle or curvature information.

Summary of Porill and Pollard IVC 1991

This section summarizes [126]. They start with edge detection and linking in two views. An initial epipolar geometry is given, but is imprecise. The goal is to optimize the epipolar geometry, which is a representation of the two-view calibration (relative pose).

The epipolar geometry is represented by the essential matrix E . First, detect edges tangent to the epipolar line (points with epipolar tangency) using the initial E . Edges with epipolar tangency are sparse since they are localized along epipolar line. The orientation of the edges is used to obtain a sparse feature set and thus reduce the correspondence ambiguity. The epipolar tangent edges are then matched in pairs, which is done by hand in this paper. The same is done for intersections of corners with epipolar lines.

If the epipolar geometry is not known, the authors provide an idea to test all possible epipolar line correspondences and positions for the epipoles to show that curve tangencies can never correspond under any epipolar geometry. In order to perform such tests efficiently, the images can be divided, using the aspect graph concept, into regions of epipole positions for which curves have stable descriptions (in this case, stable number and type of corresponding epipolar tangencies). This concept of aspect graphs for configurations of epipolar geometry with respect to curves helps justify that the iterative optimization method works if the epipoles are initialized in a correct stable region. As described below, the optimization clearly requires the initial solution to be already inside the correct stable region. In Appendix D, we explore a similar aspect graph approach to delineate valid epipolar regions. Note however, that although the aspect graph idea for this problem was proposed in this paper, it was not used in practice by the authors.

If we represent the epipolar geometry by an epipole, the distance along the curve between the new (*i.e.*, after one iteration of the optimization) and old epipolar tangent edges can be computed to the first order from the curvature of the curve, as illustrated in Figure 2.2.

The goal is to refine the calibration and the matching point coordinates so as to improve their quality as reflected in the essential constraint residual $\|xEy\|$. The free variables are the six parameters in E and the two arclength changes, one for each matching edge. An overview of the algorithm

is as follows.

- Start with a sparse correspondence of edges at epipolar tangencies (or corners) and coarse epipolar geometry
- Fix the arclengths and change E using closed form solutions for both the increments and the differential of $\|xEy\|$ with respect to E
- Fix E and change the arclengths using closed form solutions for the increments based on curvature of the curves
- Iterate on the above two steps.

At least 8 corresponding epipolar tangency points in total between two views are first computed using the coarse epipolar geometry. This number of points is necessary for using Longuet-Higgins' simple 8-point algorithm for computing the essential matrix, although 5 points would be sufficient for this task. The method refines the epipolar geometry and the initial corresponding feature points by optimizing the epipolar constraint equation $\gamma_i^{2\top} E \gamma_i^1 = 0$ for all corresponding points:

$$\min_{E, \gamma_i^1, \gamma_i^2} \sum_i \|\gamma_i^{2\top} E \gamma_i^1\|^2. \quad (2.1.1)$$

The minimization is performed by an iterative scheme, alternating updates of E keeping the points fixed, and updating the points given the new essential matrix.

Given a current solution of E , the authors derived a pair of linear equations for the changes in position of the matching feature points along their respective curves. In order to obtain those equations, we first note that when two curves have corresponding epipolar tangencies at γ_i^1 and γ_i^2 , the tangent vectors at the tangencies \mathbf{t}_i^1 and \mathbf{t}_i^2 must lie along the epipolar lines. At tangencies we thus have two extra linear conditions on E :

$$\mathbf{t}_i^{2\top} E \gamma_i^1 = \gamma_i^{2\top} E \mathbf{t}_i^1 = 0, \quad (2.1.2)$$

which we call the *epipolar tangency constraints*. In order to derive them, recall (see Appendix D) that $E\gamma_i^1$ represents the three coefficients of the equation of the epipolar line of γ_i^1 in the second image. The first two coefficients on the equation of a line $ax + by + c = 0$ represent a vector along the normal to the line. The third coefficient is irrelevant for the above equations, since the tangent \mathbf{t} was obtained by differentiating $\gamma = (\xi, \eta, 1)^\top$ and has third coordinate equal to zero. Hence, the above equations are saying that, at an epipolar tangency, the tangent to the curve has to be perpendicular to the normal of the epipolar line.

At each iteration, let E_0, γ_{i0}^k represent the initial values of the essential matrix and corresponding points. Given the updated essential matrix E , we want updated points γ_i^k satisfying the essential constraint and the epipolar tangency constraints. The updated points can be written with respect to the initial ones by a first-order Taylor approximation of the underlying curves:

$$\gamma_i^k = \gamma_{i0}^k + d\gamma_i^k, \quad k = 1, 2, \quad (2.1.3)$$

ignoring second and higher-order terms. Plugging this into the essential constraint, we have:

$$(\gamma_i^2 + d\gamma_i^2)^\top E(\gamma_i^1 + d\gamma_i^1) = 0, \quad (2.1.4)$$

to first-order. From now on, we will drop the point subscript i wherever it is clearly understood that the equations are valid for all i . Using $d\gamma^k = ds^k \mathbf{t}^k$, with s being the arclength, we have:

$$(\gamma^2 + ds^2 \mathbf{t}^2)^\top E(\gamma^1 + ds^1 \mathbf{t}^1) = 0, \quad (2.1.5)$$

that is, the increments ds^k are found by looking at what happens when we treat the curves as straight lines, locally. However, since epipolar tangencies do not change along a straight line, this equation does not constrain the increments. To see this, let us multiply-out the terms in the above equation, ignoring second-order terms:

$$\gamma^{2\top} E \gamma^1 + ds^2 \mathbf{t}^{2\top} E \gamma^1 + \gamma^{2\top} E ds^1 \mathbf{t}^1 = 0. \quad (2.1.6)$$

The last two terms are zero, by the epipolar tangency equations. Therefore, we end up with no constraint at all on the point increments (at least to first-order). What we really need is to go one order higher, that is, use the epipolar tangency equations in order to constrain these increments. In effect, we will be differentiating the epipolar tangency equations (keeping the essential matrix constant). To first-order, we can write the second epipolar tangency equation as:

$$\gamma^{2\top} E \mathbf{t}^1 = (\gamma_0^2 + d\gamma^2)^\top E(\mathbf{t}_0^1 + d\mathbf{t}^1) = 0. \quad (2.1.7)$$

Using $d\gamma^k = ds^k \mathbf{t}^k$ and $d\mathbf{t}^1 = ds \kappa_0^1 \mathbf{n}_0^1$,

$$(\gamma_0^2 + ds^2 \mathbf{t}^2)^\top E(\mathbf{t}_0^1 + ds \kappa_0^1 \mathbf{n}_0^1) = 0. \quad (2.1.8)$$

Distributing the terms and ignoring high-order quantities we have

$$\gamma_0^{2\top} E \mathbf{t}_0^1 + ds^2 \gamma_0^{2\top} E \kappa_0^1 \mathbf{n}_0^1 + ds^1 \mathbf{t}^1 E \mathbf{t}_0^1 = 0, \quad (2.1.9)$$

where we dropped the point sub-indices for simplicity. This is an equation constraining the increments ds^1 and ds^2 . There is another such equation derived from the other epipolar tangency condition $\mathbf{t}^{2\top} E \gamma^1 = 0$. Hence, we have a pair of linear equations for the changes in position along the curve, to be solved after every update of the essential matrix.

Summary of Kahl and Heyden ICCV 1998

The paper [85] proposes an analogous of the fundamental matrix to conics, in a purely algebraic approach. Given a set of conics in two views known to be in correspondence, but unknown alignment (*i.e.*, pointwise correspondence), the goal is to compute the epipolar geometry consistent with them. The basic geometric intuition is that epipolar tangencies have to correspond, thus each conic correspondence gives 2 independent constraints (2 tangencies per conic). A basic challenge is due to an

apparent chicken-and-egg problem: we need the epipolar geometry to obtain the epipolar tangencies; but the epipolar geometry must be computed from the epipolar tangencies. However, note that the paper is fully algebraic and never mentions epipolar tangencies, perhaps due to this chicken-and-egg ambiguity in defining them.

The corresponding conics can be either images of planar conics or silhouettes of quadrics. Four conics are needed to estimate the fundamental matrix, since each pair of conics gives 2 constraints, and 4 conics give 8 constraints, enough to constrain the 7 DOF in F . The problem of estimating the fundamental matrix from 5 point correspondences and 1 conic in general has 10 different solutions, obtained as the roots of a tenth degree polynomial. This is a result analogous to the 7 point correspondence case, which yields 3 different solutions in general. The link with the absolute conic is a motivator, and is integrated into the approach. If the cameras are (intrinsically) calibrated, then absolute conic is known, which gives 2 constraints on epipolar geometry just like any other conic. Thus, in the (intrinsically) calibrated situation the essential matrix can be recovered from 3 conic correspondences.

A new fundamental equation for conics is presented, in which 6 matrices appear, called the **conic fundamental matrices**. We now transcribe the basic equations. The book [69] reviews basic material on conics. If l is a dual representation of a 2D image conic, and L is a dual representation of a 3D space quadric that projects into l , we can write

$$\lambda l = PLP^\top, \quad \lambda \neq 0, \quad (2.1.10)$$

where λ is a scale factor. We can also express the symmetric matrix l as a 6-vector \tilde{l} , and L as a 10-vector \tilde{L} , and write

$$\lambda \tilde{l} = \tilde{P}\tilde{L}, \quad (2.1.11)$$

where \tilde{P} is a 6×10 matrix, whose entries are quadratic expressions in the entries of P .

Let \tilde{l}_1 and \tilde{l}_2 denote corresponding conics. Then the constraints imposed by the correspondence can be written

$$\tilde{l}_1^\top F_i \tilde{l}_2 = 0, \quad i = 1, \dots, 6, \quad (2.1.12)$$

where the matrices F_i are called **fundamental matrices for conics**, and all have rank ≤ 2 . There are three such linearly independent matrices (in the entries of F_i), something that could probably be expressed in tensor notation. However, there are only two algebraic independent constraints, in the sense that the algebraic variety defined by the 6 equations above has dimension 4. The proof of this involves using techniques for analyzing polynomial equations and Maple [31].

The numeric experiments in the paper are all synthetic, and use Equation 2.1.12 only as a residual within an iterative optimization framework. The optimization was initialized randomly, and the percentage of runs that converged was measured, for varying numbers of conics. The experiments showed this method to be extremely sensitive to the initialization, even when a large number of conics (50) is used.

Summary of Kaminski and Shashua IJCV 2004

The paper [86] extends multiple view geometry techniques to work with algebraic curves. From corresponding algebraic curves in multiple uncalibrated views, the purpose is to recover the epipolar geometry, the camera poses, and the 3D reconstruction. The problem of *how* to obtain the curve correspondences is not dealt with in this paper – this is done by hand in some experiments.

The main results of the paper are as follows:

- Provides the minimum number of corresponding algebraic curves required to solve for the epipolar geometry. This Extends Kruppa’s equations to describe the epipolar constraint of two projections of a general algebraic curve, beyond just conics.
- Studies the recovery of the homography induced by a planar space curve, given corresponding projections of this curve. This homography gives point-to-point correspondence information between views, thus being more constraining than the epipolar geometry.
- Addresses the problem of reconstructing an algebraic space curve, given corresponding projected curves. The epipolar geometry between the views is assumed known, but no alignment is given between the projected curves. If the degree of the curve is greater than 2, the reconstruction is unambiguous – just intersect the reprojection cones in 3D. If the degree is less than or equal to 2, more than two views are necessary for an unambiguous reconstruction.
- Other forms of reconstruction are shown. The dual space representation (tangent lines), and the Grassmanian-based representation. The latter requires no fitting in the image domain (although many images are required), and allows the 3D polynomial to be found from the image-based point representation of 2D curves.

Many of the proposed problems are reduced to solving systems of polynomial equations. Recent computational algebraic geometry algorithms apply [31].

Summary of Sinha *et. al.* CVPR 2004

In [140], multiple video cameras are looking at the same object. The cameras are static, and the background is controlled, so that the object silhouettes are easily segmentable. At a snapshot at $t = t_0$, we have two outer epipolar tangents at a silhouette for a given hypothesized epipole. We can use a snapshot at $t = t_1$ to get two other outer epipolar tangents for the same epipoles, therefore constraining the geometry, since three corresponding epipolar lines suffice for determining all others (see Appendix D). Figure 2.3 illustrates the idea.

However, the epipoles are unknown. Therefore, a RANSAC-like search in this 4D space is performed. The epipoles can be found by intersecting two epipolar tangency lines. Therefore, hypothesizing an epipole is the same as hypothesizing two epipolar tangency lines. The authors implement this search by first computing the convex hull of the 2D object, then computing its dual – for each point in the boundary of the convex hull, there will be a tangent line. Random sampling is done by picking two tangent lines at random, intersecting them to find a hypothetical epipole, then using

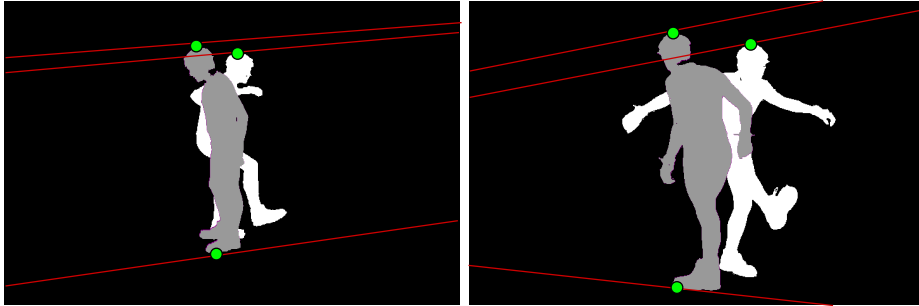


Figure 2.3: From [140]: given hypothesized epipoles, three corresponding epipolar lines are needed to solve for the complete epipolar geometry. The authors easily obtain two such correspondences at an instant $t = t_0$ (silhouette in gray) though the outermost epipolar tangencies, and a third correspondence is obtained by keeping the camera fixed but taking a snapshot at $t = t_1$ and using an outer epipolar tangency (at the white silhouette).

another video frame to get a third epipolar tangency. Now all other tangencies are predetermined (since three corresponding epipolar lines determine all others), and consensus is measured by how many frames have epipolar tangencies predicted by the hypothetical epipolar geometry. Repeat picking two tangents until maximum consensus is met, or until the probability of picking the right tangents is higher than a tolerance.

The calibration is extended for more than two views by doing every view to a central view, then finalizing with a bundle adjustment. The original paper actually does all this without knowing intrinsic parameters, obtaining a projective reconstruction, then it performs projective bundle adjustment, and it finally goes on to an auto-calibration procedure, and finally a metric bundle adjustment.

2.1.2 Infinitesimal Motion

If we measure image velocity in n pixels, $\gamma_t(1), \dots, \gamma_t(n)$, can we recover a first order differential motion model for the camera motion? Maybank was the first to show that the number of solutions, in this case, is at most 10, similar to the essential matrix situation for discrete motion [106]. In the case of curves, however, one cannot measure the image velocities at each point of the curve, since there is no epipolar geometry and the curve is localized in only one dimension. We can, however, measure the velocity at each point by tracking along the normal direction at each point of the curve. So the question arises: can a camera motion model be recovered from these measured normal velocities? The answer is, in theory, yes [48, 120], by means of a polynomial equation relating these velocities to the differential camera motion model. This is reviewed in more detail in the next section. Note, however, that this polynomial equation is corrected in the present thesis in Chapter 3, as well as extended to the case of occluding contours.

The main drawbacks of previous work on curve-based camera calibration from tracked curves in

differentiable motion are:

- Impractical implementations
- Erroneous formulas
- Occluding contours are not treated

We now review the paper [48] and related publications in further detail.

Summary of Faugeras and Papadopoulos IJCV 1993

The paper [48] is the main reference for the study of infinitesimal structure and motion recovery for curves (except occluding contours). Experience with implementing the theory in [48] appeared as a technical report [121], where one can find stability experiments on real data, and also in [122]. The latter paper also proposes and explores visibility constraints for curves. There is a version of [48] that appeared in Mundy’s geometric invariance book [114, ch.16]. The definitive reference, however, is Papadopoulos’ PhD thesis [120], which contains the most up to date formulas and the explicit polynomial equations not shown in the original papers.

Faugeras and Papadopoulos showed that for non-rigid 3D curves seen from an uncalibrated monocular video, reconstruction is dependent on tangential velocity, which is not observable from the geometry. Multiple views are needed in this case, as explored in more recent work [22, 37]. For rigid 3D curves, the reconstruction is in principle possible: the normal velocity at each curve point allows for the recovery of a 2^{nd} order motion model of the curve (or, alternatively, the camera motion model assuming camera is moving and the curve is fixed), as well as the curve shape. Specifically, each point’s second order spatio-temporal derivative gives a polynomial equation, referred to as the $L1$ equation, in the motion model parameters (rotation velocity, translation velocity, and their derivatives). Although the original paper [48] claims there are two polynomial equations per point, in [121], they realize that the two polynomial equations are redundant – one of them can be expressed as a linear combination of the other and its first time derivative. The final polynomial equation can be found in [120], but we found out it also has an error due to a missing term which we provide in Chapter 3 of this thesis.

Even though most of the paper considers a monocular sequence of views of a moving curve, there is a section towards the end regarding a binocular stereo rig observing a rigidly moving curve (or, similarly, a moving stereo rig observing a fixed curve). The case of a stereo rig assumes both cameras fully calibrated both intrinsically and extrinsically relative to each other, and the goal is to disambiguate matches along epipolar lines. Let there be a curve in the left image for which we wish to find a matching curve among many candidates in the right image. The proposed constraint is basically that the wrong correspondences cannot be modeled by a 3D curve moving rigidly.

The authors derive motion constraints from curve tangents. At each time instant, a 3D tangent can be hypothesized/reconstructed from two putatively corresponding image tangents by a formula we review in Chapter 3. Across different time instants, for each view there is another formula to

hypothesize/reconstruct a 3D tangent given tracked tangent in at least two frames. We review this formula in the end of Chapter 3. Combining the equations for the 3D tangent, they obtain two equations describing the projection of the angular velocity in the plane the plane normal to the 3D tangent (equation 70 of the paper). The authors give an idea on how to use these constraints in practice, although no experiments are performed in the paper.

The main contributions of the paper are: (i) the relation between 3D motion and curve geometry to spatio-temporal measurable geometry of the image curves; (ii) the recovery of 3D motion and depth from observed image curve geometry; and (iii) the use of a rigidity constraint to disambiguate stereo correspondence when the scene is moving.

The main drawback of the paper, according to other authors, is that the high-order derivatives are difficult to compute.

2.2 Curves in Stereo

2.2.1 Multiview Stereo Using Curves

These papers are concerned with the following problem: given a system of cameras that are well-calibrated both intrinsically and extrinsically relative to each other, reconstruct the scene. The most common class of methods in this category are those making use of silhouette curves [14, 74, 91, 92]. Typically there is only a single object per scene, and the acquisition is controlled so that the outer silhouette curves can be easily extracted. Once the silhouettes are extracted in each view, they are backprojected in space forming a 3D visual hull. This reconstruction provides a rough surface model for the object. The visual hull is then used as a starting point for a surface-based evolution that optimizes photometric constraints while keeping the surface constrained to have the observed silhouettes. Figure 1.8 illustrates this process. Although these methods make good use of curves in specific domains, their main drawbacks are:

- Controlled acquisition: require well-segmentable silhouettes, and usually require precisely calibrated cameras.
- Can't handle multiple objects, limiting their applicability to general scenes.

Another type of work was developed by Wu and Yu [159], where a multiview reconstruction using curves is performed manually in a CAD fashion. We review this paper next, as it forms a main previous work for our own reconstruction system described in Chapter 5.

Summary of Wu and Yu VC 2005

The paper [159] describes an interactive system for multiview 3D reconstruction using curves. For each view, the user marks vertices, curves connecting vertices, and surface patches bound by a chain of curves, as well as their correspondences across the views. The intrinsic parameters for each view

are known. The output of the system is a 3D texturized surface model with important curves and the camera poses for each view.

From the point correspondences that the user provides, the cameras are found by stitching together views and solving for the essential matrix, followed by bundle adjustment of the 3D points and cameras. Once the system is fully calibrated, the curves are reconstructed using dynamic programming followed by a bundle adjustment. This generates a 3D curved wireframe model. The surface patches bounded by chains of curves are then reconstructed by two possible methods: (i) dense stereo in case there is enough texture in the region, or (ii) thin-plate spline interpolation of the wireframe and point reconstruction, in case there is little texture in the region. For display purposes, a triangular mesh is generated from the continuous model, and image-based texture-mapping is performed.

The reasons for using curves are that: (i) Man-made scenes do not have much texture, but have many curves (*e.g.*, a car, a printer, or a sofa). For these types of scenes, the curves can describe the object fairly well; and (ii) curves can be easily specified by the user.

No smooth occluding contours are supported. All occluding contours are approximated by a hard dihedral edge. Curves are mainly used within the reconstruction problem: given corresponding curves with known corresponding endpoints and no linking problems, find the alignment between them (*i.e.*, the pointwise correspondence) and the 3D reconstruction. The input curves are represented by linearly interpolated pixel samples connecting the endpoints.

Because of the epipolar constraint, solving the point mapping function between two image curves seems trivial at first. For every point on the first curve, the intersection between its epipolar line and the second curve is the corresponding point in the second curve. However, this is true only when there is exactly one such intersection – no more, no less. In reality, there might be zero or multiple such intersections, due to the shape of the curves and uncertainty in the camera poses. In the worst case, the image curve is almost straight but parallel to the epipolar line to cause huge amount of uncertainty in the location of the intersection.

The multiple view curve reconstruction is formulated as follows. Let an image curve $\gamma(s)$ be parametrized by $s \in [a, b]$. When there are m corresponding image curves, one for each view, we have $\gamma_i(s_i)$, $i = 0, \dots, m - 1$, each of which with a distinct parameter $s_i \in [a_i, b_i]$. The correspondence between the endpoints of these m curves are known. Pick γ_0 as the base curve and assume that the endpoint $\gamma_0(a_0)$ corresponds to $\gamma_i(a_i)$, $i = 1, \dots, m - 1$. Thus, obtaining point correspondences among these m curves is equivalent to solving $m - 1$ mappings $\sigma_i(s_0)$, $i = 1, \dots, m - 1$, each of which is a continuous and monotonically increasing function that maps $[a_0, b_0]$ to $[a_i, b_i]$. For closed curves, as long as there are at least two point features on each of them corresponding to one another, each closed curve can be broken into two or more open curves.

The relative rotations and translations between the i th and the j th camera frames are respectively denoted by R_{ij} , T_{ij} , $i, j = 0, \dots, m - 1$. The epipolar constraint between corresponding points on the i th and j th curves requires that

$$\gamma_j(\sigma_j(s_0))^\top [T_{ij}]_\times R_{ij} \gamma_i(\sigma_i(s_0)) = 0, \quad s_0 \in [a_0, b_0]. \quad (2.2.1)$$

Thus, the desired mappings should minimize some kind of geometric distance between corresponding epipolar lines. Furthermore, to guarantee that $\sigma(s)$ is a monotonically increasing one-to-one mapping, $\sigma(s) \leq \sigma(s')$ must be held for arbitrary $s \in [a, b]$ and $s' \in [a, b]$ such that $s < s'$. To incorporate these considerations, the minimization problem should be formulated as

$$\begin{aligned} \min_{\sigma_i, 1 \leq i \leq m-1} \sum_{ij, i < j} \int_{a_0}^{b_0} \{ \text{dist}^2[\gamma_j(\sigma_j(s)), el_{ij}(\gamma_i(\sigma_i(s)))] + \text{dist}^2[\gamma_i(\sigma_i(s)), el_{ji}(\gamma_j(\sigma_j(s)))] \} ds \\ + \lambda \sum_i \int_{a_0}^{b_0} \int_s^{b_0} \max\{\sigma_i(s) - \sigma_i(s'), 0\}^2 ds' ds, \end{aligned} \quad (2.2.2)$$

where $el_{ij}(p)$ denotes the epipolar line on view j of point p on view i . The first term enforces the epipolar constraint, and the second term enforces that $\sigma_i(s)$ is a 1-1 mapping. The parameter λ indicates the relative importance of the two terms, and is set to a large value such as 10^3 .

In practice, the implementation considers curves with subpixel accuracy defined as the piecewise linear interpolation of a discrete set of pixels. The longest image curve serves as the 2D parametrization of the 3D curve, and is denoted as $\gamma_0(s_0)$.

Both the quasi-Newton and conjugate gradient descent methods can effectively minimize the discretized cost function. However, a reasonable initialization is required before the nonlinear optimization. The proposed method initializes the mappings using dynamic programming, which is suitable for order-preserving one-dimensional mappings. Each $\sigma_i(s)$ is initialized independently using only two curves (γ_0 and γ_i) and adopt the discrete version of the first term in (2.2.2) as the cost function for dynamic programming. The one-to-one mapping is enforced as a hard constraint, since only order-preserving mappings are admissible.

More specifically, the authors represent each curve γ_i as a discrete set of pixels p_i^k , $k = 0, \dots, n_i$, where n_i is the number of pixels in the on the curve. Dynamic programming recursively computes the overall mapping cost. The cumulative cost between a pair of pixels on the two curves is defined as:

$$C_{dp}(p_0^k, p_i^l) = D(p_0^k, p_i^l) + \min_{r \in S_{kl}} C_{dp}(p_0^{k-1}, p_i^r), \quad (2.2.3)$$

where

$$D(p_0^k, p_i^l) = \text{dist}^2[p_0^k, el_{i0}(p_i^l)] + \text{dist}^2[p_i^l, el_{0i}(p_0^k)]$$

and S_{kl} contains all admissible values of r under the condition that p_0^k matches p_i^l .

Once the alignment functions are obtained, then one can triangulate discrete set of corresponding points $s_0 \in [a_0, b_0]$, followed by bundle adjustment to refine the 3D positions. The reconstructed 3D points are essentially discrete ordered vertices. If a parametrized smooth 3D curve reconstruction is required by the user, the authors use subdivision curves, optimizing their control points such that a discrete set of samples project as closely as possible to the image curves.

2.2.2 Binocular and Trinocular Stereo Using Curves

The goal of these papers is to find edge or curve correspondences across two or three well-calibrated views, and reconstruct the curves based on these correspondences. The key bottleneck is the disambiguation of correspondences.

Initial work in this area consists in binocular stereo of curves [3,9,17,42,108,115,123]. However, in the case of two views, real geometric constraints cannot be applied, due to the nature of curves being localized in only one direction. Accordingly, all these methods are based on some sort of heuristic similarity between the two views. This limits these methods to short baseline, a tight depth range, and to scenes without too many depth discontinuities or branching structures. The method by Arnold and Binford [3], for example, proposed the use of curve tangents to disambiguate correspondences. The idea is that the tangents tend to be similar at corresponding points of two matching curves. This, however, is only true for very short baseline and limited depth range. In reality, any two tangent directions are possible, as we show in Chapter 3.

The idea that the differential geometry of curves can be used to correlate structure in three images was presented in the work of Ayache and Lustman [8], who proposed a trinocular constraint for matching line segments arising from edge linking. The main idea is that a 3D point and its tangent reconstructed from a pair of potentially corresponding points and tangents in two views determine a point and tangent in a third view, which can be compared to observations; see also [70,136,141]. Robert and Faugeras [131] extended Ayache’s method of transferring points and tangents from two views to a third to include curvature: 3D curvature and normal can be reconstructed from 2D curvatures at two views, which in turn determine the curvature in a third view. This leads to improved precision and density in the reconstruction since curvature provides an additional constraint and reinforces figural continuity in propagating strong hypotheses to neighboring curve samples. This allows then to discard the aforementioned use of heuristics such as binocular similarity or the ordering constraint [117]. Schmid and Zisserman [133] also derived a formula for transferring curvatures from two views to a third, using a projective geometry formalism in which the osculating circle is transferred as a conic. The advantage of their approach is that the cameras don’t have to be metrically calibrated – only the projective aspects of epipolar geometry are required.

Li and Zucker [94,95] derived formulas for the curvature of a projected curve from the curvature of a 3D space curve. They also derived a system of linear equations for reconstructing 3D curvature from 2D, as previously done by Faugeras [131], but with a different proof. Their stereo method assesses the compatibility of *two* neighboring point-tangent-curvature matches according to a cost, which is then minimized through relaxation labeling. While tangents and curvatures can be reconstructed, torsion cannot be constrained. Therefore their process minimizes the torsion of the resulting 3D curve, assuming real-world curves tend to have low variation.

Another class of papers deals with stereo reconstruction using parametric models [11,30,34,34,81,84,93,128,132,135,153,160].

The main drawbacks of previous work on curve-based binocular and trinocular stereo are:

- The use of biased heuristics in binocular stereo

- The requirement of very precise calibration

Summary of Schmid and Zisserman 2000

This section reviews [133]. The paper deals with the problem of finding correspondence between curve segments when 2 or 3 images from unknown cameras are available. The Fundamental matrix or trifocal tensor is easily computable from interest points + RANSAC, but the intrinsic parameters are not necessarily known. The classic approach is to discard these interest points and only use the camera geometry in a subsequent stereo matching stage to disambiguate point correspondences along epipolar lines. For 2 views, the standard method of computing point correspondences is using the epipolar constraint, which limits the candidates to be in a narrow region, and the use of photometric normalized cross correlation to further constrain the choice.

A central question answered here is: are there any additional constraints when not points but point-tangents (edges) or in general curve geometry is available? Let us analyze the binocular and the trinocular cases.

The binocular case. The correspondence search from photometric correlation can be significantly reduced by using curve geometry (not arising from occluding contours) in a small neighborhood. Consider a first-order (planar) approximation to the surface of the object on which the 3D curve lies. The correspondence between the projections of this plane in two views is governed by a homography of 8 free parameters. It is known that the fundamental matrix, which captures the epipolar geometry, gives 5 constraints, reducing the homography to 3DOF. Thus, the fact that a point is matched unto a point on the epipolar line is already taken into account. Given the candidate matches of a curve fragment along the epipolar line, each pair of matches gives 2 constraints on the unknown variables: as the curves have to map to each other, (i) the position along the epipolar lines defines one variable, since the 3D plane is constrained to pass through a 3D point reconstructed from this information (up to 3D homography); and (ii) the matching tangents provide an additional constraint, since the 3D plane also has to pass through the 3D tangent reconstructed from this information (up to 3D homography). This leaves a 1-parameter family of solutions which can be optimized by the degree of photometric correlation.

Among the remaining 1-parameter family of solutions, the authors propose that, instead of optimizing photometric correlation, one can just pick the plane that coincides with the osculating plane of the curve, which can be determined by using curvatures of the matching points. This is just a heuristic – there is no guarantee that this is the tangent plane to the underlying surface, even if the curve is planar. However, empirically the authors state that this solution was good enough to rank-order the matches even in the above cases.

The question of how to get the osculating plane (and its associated homography) motivated a result which is of more general interest. It is known that from the position and curvature of corresponding edgels in two views, it is possible to estimate the osculating plane of the curve uniquely, if the imaging calibration was fully known [46], as shown in Chapter 3. If only the fundamental

matrix is known, the osculating plane can only be recovered up to a projective ambiguity, but its associated homography can be fully recovered, and this is studied in this paper.

The trinocular case. The authors also propose a method for transferring the curvature from two views onto a third, given the trifocal tensor (the full calibration is not needed). The interpretation of the given formula is to use the reconstructed plane from two views (known up to 3D projective ambiguity), and use that to get a homography relating the first two views to the third one (this homography being independent of the intrinsic parameters). This homography then defines a pointwise correspondence between any of the first two views to the third one. Faugeras and Robert [49] were the first to propose the transfer of curvature from two uncalibrated views to a third one, but they used pairwise fundamental matrices. The improvement over Faugeras’ method proposed by Schmid and Zisserman is numerical stability and immunity to errors in intersecting epipolar lines that are collinear or nearly collinear. The trifocal tensor handles these cases without any imprecision.

Steps of practical system.

- Start with edge detection and linking in two or three views, and attempt to match whole fragments of curves
- At a first stage, consider only curves that lie in the beam of epipolar lines of each other and have consistent epipolar tangencies.
- At a second stage, attempt to find corresponding fragments of putative corresponding curves by integrating edgel-to-edgel costs within possibly matching fragments.
- Using only the differential geometry of the curves provides constraints by transferring putative edgel matches to a third view. This works even when the intrinsic parameters are unknown, but ambiguity remains
- Further constraint can be found using appearance.
- For each putative match, find a correlation score by searching on a 1-parameter space if the curvatures of the matches are low (only tangents are used), or using no search at all (osculating plane) if the curvatures are above a threshold.
- A winner-take-all scheme picks the final matches

2.3 Occluding Contours for Surface Reconstruction in Video

As a matter of terminology, occluding contours in 3D are also called *contour generators*, while projected contour generators are called *apparent contours*. The papers in this category have the goal of reconstructing a local surface model (typically second-order) given the camera models for each frame of a video sequence, together with observed apparent contours. The way an apparent

contour deforms in the video as the camera moves can provide a local surface model as long as at least three frames are used. Usually, a pointwise correspondence between the apparent contours in each frame is established by intersecting corresponding epipolar lines with the curves. There are other possible parametrizations, such as going along the normal to the curve to obtain the corresponding curve point in another frame, although these are much less common than the epipolar parametrization.

The main drawbacks of the literature in differential surface reconstruction from occluding contours in calibrated video is are:

- Highly controlled acquisition, requiring the curves to be easy to segment and track in the frames
- Lack of unified theory with other types of contours. Most papers deal only with occluding contours, disregarding fixed curves, with exception of a single paper [98].

We now summarize the main papers in the literature.

Summary of Koenderink Occluding Perception 1984

This section reviews [88]. The main result of this paper is that the sign of the curvature of the apparent contour equals the sign of the Gaussian curvature at the corresponding point of the contour generator. Thus, parabolic points on the a surface correspond to inflections in the apparent contour. This result is also valid for perspective projection. The conclusions of the paper come from two mathematical results which will be derived below. For a brief overview, the reader can skip the proofs.

Theorem 2.3.1. *The Gaussian curvature \mathcal{K} of the surface \mathcal{M} , the curvature \tilde{K}^γ of the apparent contour under orthographic projection, and the radial curvature K^r , the normal curvature of \mathcal{M} along line of sight, are related by:*

$$\mathcal{K} = K^r \cdot \tilde{\kappa}^\gamma. \quad (2.3.1)$$

Proof. Consider local object coordinates $(\bar{x}, \bar{y}, \bar{z})$ defined as:

- \bar{x} : along visual ray,
- \bar{z} : along normal to surface \mathcal{M} ,
- \bar{y} : normal to both \bar{x} and \bar{z} directions.

Since we have orthographic projection, the $\bar{y} - \bar{z}$ plane is parallel to image plane. We can write the surface as $\bar{z}(\bar{x}, \bar{y})$, expanding to second order:

$$\bar{z} = \frac{1}{2}(a\bar{x}^2 + 2b\bar{x}\bar{y} + c\bar{y}^2) + \text{h.o.t.} \quad (2.3.2)$$

The contour generator is defined by the condition that the normal $\bar{\mathbf{N}}$ (written in local coordinates) is orthogonal to the viewing direction \mathbf{e}_1 along \bar{x} :

$$\mathbf{e}_1 \cdot \bar{\mathbf{N}} = 0 \quad (2.3.3)$$

or

$$\mathbf{e}_1 \cdot (-\bar{z}_x, -\bar{z}_y, 1) = -\bar{z}_x = 0. \quad (2.3.4)$$

Thus, the equation

$$\bar{z}_x = \frac{d\bar{z}}{d\bar{x}} = a\bar{x} + b\bar{y} + \text{h.o.t.} = 0 \quad (2.3.5)$$

defines the contour generator. The solution of this equation is a set of (\bar{x}, \bar{y}) such that $(\bar{x}, \bar{y}, \bar{z})$ is in the contour generator.

It follows that the contour generator is not orthogonal to the \bar{x} direction, but is *conjugated* to it. The angle between these directions is given by $\tan^{-1}(-a/b)$. From (2.3.5), we have $a\bar{x} = -b\bar{y}$, so

$$\bar{x} = -\frac{b\bar{y}}{a}. \quad (2.3.6)$$

Plugging into (2.3.2),

$$\bar{z} = \frac{1}{2} \frac{ac - b^2}{a} \bar{y}^2 = \frac{1}{2} \tilde{K}^\gamma \bar{y}^2, \quad (2.3.7)$$

which is the equation of the apparent contour to second order, giving the curvature \tilde{K}^γ of the apparent contour under orthographic projection as:

$$\tilde{K}^\gamma = \frac{ac - b^2}{a}. \quad (2.3.8)$$

The radial plane is the $\bar{x} - \bar{z}$ plane ($\bar{y} = 0$), so that, neglecting third and higher order terms:

$$\bar{z} = \frac{1}{2} K^r, \quad (2.3.9)$$

where

$$K^r = a. \quad (2.3.10)$$

Using a formula for the Gaussian curvature of surface as a function $\bar{z}(\bar{x}, \bar{y})$ (Equation (A.3.39)), and computing the required derivatives from (2.3.2), we obtain:

$$\mathcal{K} = ac - b^2 = K^r \cdot \tilde{\kappa}^\gamma. \quad (2.3.11)$$

■

The following theorem gives a similar result for perspective projection.

Theorem 2.3.2. *The Gaussian curvature \mathcal{K} , the radial curvature K^r at a point Γ of a contour generator, and the geodesic image curvature κ^{app} are related by:*

$$\kappa^{app} = \hat{\rho} \frac{\mathcal{K}}{K^r}, \quad (2.3.12)$$

or

$$\mathcal{K} = \frac{\kappa^{app} \cdot K^r}{\hat{\rho}}, \quad (2.3.13)$$

assuming perspective projection, and where $\hat{\rho}$ is the depth of the contour generator Γ along the visual direction $\hat{\gamma}$.

Proof. The geodesic image curvature κ^{app} is defined as follows. Consider a unit sphere around the camera center, *i.e.*, the manifold of visual directions. The contour generator projects to an apparent contour on such a sphere. The geodesic image curvature is the curvature of the orthogonal projection of such a contour onto the tangent plane to the viewing sphere at the considered point. This is *not* the same as curvature of the apparent contour for a planar image. However, according to Koenderink, inferences of curvature sign from such formulas should apply to all cases.

Let us consider the *osculating cone* of visual rays, defined as the second-order cone that must approximate the real cone of tangent rays to the surface \mathcal{M} emanating from \mathbf{c} . All osculating cones envelop a sphere, called the *Mannheim sphere* [88, 89]. For $\hat{\rho} \rightarrow \infty$, the osculating cone degenerates to a cylinder with radius \tilde{K}^γ , which is also the radius of the Mannheim sphere. The cone also cuts the unit image sphere in a circle of visual directions whose geodesic curvature is κ^{app} . From simple trigonometry, we have

$$\kappa^{app} = \cot \alpha \tag{2.3.14}$$

$$= \hat{\rho} \tilde{\kappa}^\gamma. \tag{2.3.15}$$

Now, using Theorem 2.3.1, we can substitute K^r to get the desired result. ■

The curvature K^r always has the same sign along a contour generator because the visual ray can not run through the material of the surface, which is assumed to be opaque. Hence, from the formula for \mathcal{K} in (2.3.12), we conclude that the sign of the Gaussian curvature equals the sign of the curvature at the apparent contour. Inflections on the apparent contour correspond to parabolic points in the surface. Also, if the apparent contour is convex, then the 3D point is elliptic, and if the apparent contour is concave, the 3D surface point is hyperbolic.

Summary of Giblin and Weiss ICCV 1987

This section reviews [60]. This is the pioneering paper of silhouette-based reconstruction from continuous viewer motion. Even though the paper was surpassed by later works, it has a strong historical value.

The method exploited multiple silhouettes to reconstruct shape under continuous camera motion. The camera motion is restricted to be planar, and the projection is orthographic. Throughout the paper, only synthetic examples are given, with no real-life application. A pioneering idea was that a smooth surface is the envelope of its tangent planes obtained from the apparent contours. They reduced this problem to one of computing the envelope of tangent lines in a plane. The authors also reconstructed depth, Gaussian and mean curvatures along occluding contours.

Summary of Cipolla Blake IJCV 1992

This section reviews [24]. This paper is the main source of basic results of occluding contours under continuous viewer movement. Cipolla's PhD thesis [25] covers this paper and contains additional details, while most of Cipolla and Giblin's book [27] presents these results as well, with some

complementary theoretical details, but with less experiments than the thesis and the paper.

The main contributions of the paper are outlined below:

- Extended the preliminary work of [60] (which we outlined in Section 2.3) to perspective projection, a real-world application, and detailed implementation.
- Introduced the epipolar parametrization.
- Studied Parallax-based measurements that can be more robustly computed than direct measurements of position and its first and second-order spatio-temporal derivatives. The idea is to take spatial derivatives of measurements instead of isolated ones, so that errors due to global effects tend to cancel-out.
- Provided error and sensitivity analysis for the proposed formulas
- Devised a simple application to discriminate viewpoint-dependent and viewpoint-independent contours

The paper uses spherical images instead of planar images, arguing that this is a powerful analytical tool that simplified the process of extending the theory of [60] to perspective projection. In this thesis, we translate the results to planar perspective.

The authors derived a depth formula for differential motion and showed it to be independent of the surface curvature.

Conjugacy of Contour Generator and View Direction

The viewline in the direction of γ is tangent to the surface \mathcal{M} at the contour generator Γ ; this is what defines the contour generator. On the other hand, the tangent Γ_s^w to the the contour generator at the point Γ , being a curve of \mathcal{M} , has to be tangent to the surface as well. Thus, both the vectors are in the tangent plane. An interesting discussion comes from the following question. Given the viewing direction $\gamma \in T_\Gamma(\mathcal{M})$ (*i.e.*, in the tangent plane to \mathcal{M} at Γ), what is the direction Γ_s^w along the contour generator? These directions are shown in the paper not to be orthogonal, but are instead said to be *conjugate*. Conjugacy is a type of generalized orthogonality and is invariant to projective transformations, as reviewed in Section A.3.1. These results are summarized in the following theorem.

Theorem 2.3.3. *Consider a point Γ of the surface \mathcal{M} lying on a contour generator relative to a camera center c . The visual direction $\hat{\gamma}$ and the tangent direction to the contour generator \mathbf{T} are conjugate directions, *i.e.*,*

$$d\mathbf{N}(\mathbf{T}) \cdot \hat{\gamma} = \mathbf{T} \cdot d\mathbf{N}(\hat{\gamma}) = 0. \quad (2.3.16)$$

Proof. Using the epipolar parametrization $\Gamma^w(s, t)$, the vector Γ_s^w is along the contour generator tangent, and Γ_t^w is along the visual direction, as introduced in Section 3.2. The condition for conjugacy using $d\mathbf{N}(\mathbf{T}) = \mathbf{N}_s$ translates to

$$(\Gamma^w - \mathbf{c}) \cdot \mathbf{N}_s^w = 0, \quad (2.3.17)$$

where $\mathbf{N}_s^w = \frac{\partial}{\partial s} \mathbf{N}^w \circ \Gamma^w(s, t)$. Differentiating $(\Gamma^w - \mathbf{c}) \cdot \mathbf{N}^w = 0$ with respect to s , we get

$$\Gamma_s^w \cdot \mathbf{N} + (\Gamma^w - \mathbf{c}) \cdot \mathbf{N}_s = 0. \quad (2.3.18)$$

Since Γ_s^w is in the tangent plane, we have $\mathbf{N}^w \cdot \Gamma_s^w = 0$, so that

$$(\Gamma^w - \mathbf{c}) \cdot \mathbf{N}_s^w = 0. \quad (2.3.19)$$

■

Corollary 2.3.4. *For the epipolar parametrization, the tangent vectors $\{\Gamma_s^w, \Gamma_t^w\}$ are conjugate. Therefore, the matrices of the first and second fundamental forms in the unit basis along these vectors are given by:*

$$\mathbf{I} = \begin{bmatrix} 1 & \cos \theta \\ \cos \theta & 1 \end{bmatrix}, \quad \mathbf{II} = \begin{bmatrix} K^s & 0 \\ 0 & K^t \end{bmatrix}, \quad (2.3.20)$$

where: θ is the angle between Γ_s^w (along the visual direction) and Γ_t^w (along the contour generator tangential direction); K^t is the normal curvature of the t -parameter curve $\Gamma^w(t) = \Gamma^w(s_0, t)$, which is the curvature of the normal section at the point Γ in the visual direction; and K^s is the normal curvature of the contour generator $\Gamma^w(s) = \Gamma^w(s, t_0)$.

Proof.

$$\mathbf{I} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \hat{\Gamma}_s^w \cdot \hat{\Gamma}_s^w & \hat{\Gamma}_s^w \cdot \hat{\Gamma}_t^w \\ \hat{\Gamma}_s^w \cdot \hat{\Gamma}_t^w & \hat{\Gamma}_t^w \cdot \hat{\Gamma}_t^w \end{bmatrix}, \quad (2.3.21)$$

where $\hat{\Gamma}_s^w = \Gamma_s^w / \|\Gamma_s^w\|$ and $\hat{\Gamma}_t^w = \Gamma_t^w / \|\Gamma_t^w\|$, since we are taking a unit basis along Γ_s^w and Γ_t^w . Therefore, $\Gamma_s^w \cdot \Gamma_s^w = \Gamma_t^w \cdot \Gamma_t^w = 1$ and $\Gamma_s^w \cdot \Gamma_t^w = \cos \theta$.

The second fundamental form, in the same basis, can be written as:

$$\mathbf{II} = \begin{bmatrix} e & f \\ f & g \end{bmatrix} = \begin{bmatrix} \hat{\Gamma}_s^w \cdot d\mathbf{N}(\hat{\Gamma}_s^w) & \hat{\Gamma}_s^w \cdot d\mathbf{N}(\hat{\Gamma}_t^w) \\ \hat{\Gamma}_s^w \cdot d\mathbf{N}(\hat{\Gamma}_t^w) & \hat{\Gamma}_t^w \cdot d\mathbf{N}(\hat{\Gamma}_t^w) \end{bmatrix}. \quad (2.3.22)$$

Since we know that $\hat{\Gamma}_t^w$ and $\hat{\Gamma}_s^w$ are along conjugate directions, then $f = \hat{\Gamma}_s^w \cdot d\mathbf{N}(\hat{\Gamma}_t^w) = 0$ by definition. Since we are taking unit basis vectors, the off-diagonal elements can be written as $e = II(\hat{\Gamma}_s^w, \hat{\Gamma}_s^w) = K^s$ and $f = II(\hat{\Gamma}_t^w, \hat{\Gamma}_t^w) = K^t$. ■

From the previous corollary, the first and second fundamental forms are fully determined by the unknowns θ , K^s and K^t . These can be computed from viewer motion up to second-order. The formulas for these entities become simpler if we use spherical image projection, *i.e.*, $\hat{\gamma} = \gamma / \|\gamma\|$ as our image point, and $\Gamma = \hat{\rho} \hat{\gamma}$ as the projection equation. The following proposition is useful to convert expressions for θ , K^s , and K^t from spherical image projection to planar image projection. It is given in this thesis but not in the original papers by Cipolla and Giblin.

Proposition 2.3.5. (Relating planar and spherical image measurements) The depths $\hat{\rho}$ and ρ are related by:

$$\hat{\rho} = \rho \|\gamma\|, \quad (2.3.23)$$

and their derivatives $\hat{\rho}_s$ and ρ_s are related by:

$$\hat{\rho}_s = \rho_s \|\gamma\| + \rho \frac{\gamma}{\|\gamma\|} \cdot \gamma_s \quad (2.3.24)$$

Given a measurement of γ_s in planar image projection, the corresponding measurement $\hat{\gamma}_s$ in spherical image projection is given by:

$$\hat{\gamma}_s = \frac{\|\gamma\|^2 \gamma_s - (\gamma \cdot \gamma_s) \gamma}{\|\gamma\|^3}. \quad (2.3.25)$$

Proof. The first equation comes from $\mathbf{\Gamma} = \hat{\rho} \hat{\gamma} = \rho \gamma$. The second comes from differentiating the first one with respect to s :

$$\hat{\rho}_s = \rho_s \|\gamma\| + \rho \frac{d}{ds} \|\gamma\|, \quad (2.3.26)$$

where

$$\frac{d}{ds} \|\gamma\| = \frac{d}{ds} (\gamma \cdot \gamma)^{1/2} = \frac{\gamma}{\|\gamma\|} \cdot \gamma_s. \quad (2.3.27)$$

The third equation of the proposition is obtained from

$$\hat{\gamma}_s = \frac{d}{ds} \left(\frac{\gamma}{\|\gamma\|} \right) = \frac{1}{\|\gamma\|^2} \left(\gamma_s \|\gamma\| - \gamma \frac{d}{ds} \|\gamma\| \right) \quad (2.3.28)$$

and using (2.3.27). ■

We can now translate any formula that is based on spherical image projection, to one based on planar image projection.

Theorem 2.3.6. The angle θ between the viewing direction γ and the tangent to the contour generator $\mathbf{\Gamma}_s$ is given by:

$$\tan \theta = \frac{\hat{\rho} \|\hat{\gamma}_s\|}{\hat{\rho}_s}, \quad (2.3.29)$$

where $\hat{\gamma}$, $\hat{\rho}$, and their derivatives are given by Proposition 2.3.5.

Proof. Differentiating $\mathbf{\Gamma} = \hat{\rho} \hat{\gamma}$ with respect to s

$$\mathbf{\Gamma}_s = \hat{\rho}_s \hat{\gamma} + \hat{\rho} \hat{\gamma}_s, \quad (2.3.30)$$

and taking the dot product with the viewing direction $\hat{\gamma}$, we have:

$$\mathbf{\Gamma}_s \cdot \hat{\gamma} = \hat{\rho}_s, \quad (2.3.31)$$

where we used $\hat{\gamma}_s \cdot \hat{\gamma} = 0$ since $\hat{\gamma} \cdot \hat{\gamma} = 1$. We then have

$$\cos^2 \theta = \frac{(\mathbf{\Gamma}_s \cdot \hat{\gamma})^2}{\|\mathbf{\Gamma}_s\|^2} = \frac{\hat{\rho}_s^2}{\hat{\rho}_s^2 + \hat{\rho}^2 \|\hat{\gamma}_s\|^2}. \quad (2.3.32)$$

This is related to the tangent of θ by

$$\tan^2 \theta = \frac{1 - \cos^2 \theta}{\cos^2 \theta} = \frac{1}{\cos^2 \theta} - 1 = \frac{\hat{\rho}_s^2 + \hat{\rho}^2 \|\hat{\gamma}_s\|^2}{\hat{\rho}_s^2} - 1, \quad (2.3.33)$$

which gives the desired formula. ■

The required derivative of depth in Theorem 2.3.6, ρ_s , is obtained in [25] by numerically differentiating the reconstructed ρ for neighboring points.

Theorem 2.3.7. *The curvature K^s of the surface \mathcal{M} in the direction of the contour generator $\mathbf{\Gamma}$ and the transverse curvature K^t , which is the sectional curvature in the direction of the visual ray $\hat{\gamma}$, are given by:*

$$K^t = \frac{\mathbf{c}_t \cdot \mathbf{N}^w}{\hat{\rho}(\mathbf{c}_t \cdot \hat{\gamma} + \hat{\rho}_t)} \quad (t = 0) \quad (2.3.34)$$

$$K^s = \frac{\kappa^{app} \sin^2 \theta}{\hat{\rho}}, \quad (2.3.35)$$

where κ^{app} is the curvature of the apparent contour:

$$\kappa^{app} = \frac{\hat{\gamma}_{ss}^w \cdot \mathbf{N}^w}{\|\hat{\gamma}_s^w\|^2}. \quad (2.3.36)$$

From θ , K^t , K^s , we can write the formulas for the usual curvatures of the surface:

$$\mathcal{K} = \frac{\kappa^{app} K^t}{\hat{\rho}} \quad (2.3.37)$$

$$\mathcal{H} = \frac{1}{2} \left[\frac{\kappa^{app}}{\hat{\rho}} + K^t \csc^2 \theta \right] \quad (2.3.38)$$

$$\kappa_{1,2} = \mathcal{H} \pm \sqrt{\mathcal{H}^2 - \mathcal{K}} \quad (2.3.39)$$

The paper [24, p. 92] also provides a very big formula giving ρ_t in terms of γ_{tt}^w , \mathbf{c}_{tt} , and Ω_t .

Parallax-based measurements (outline). The idea is that formulas involving spatial derivatives of measurements are more robust than formulas for pointwise measurements, since they tend to cancel-out common uncertainties. If we write the velocity formula 3.5.14 for two nearby points and subtract them, we get:

$$\hat{\gamma}_{t1} - \hat{\gamma}_{t2} \rightarrow [(\mathbf{c}_t \times \hat{\gamma}) \times \hat{\gamma}] \left(\frac{1}{\hat{\rho}_2} - \frac{1}{\hat{\rho}_1} \right) \quad (2.3.40)$$

This idea is due to [100], and Cipolla and Blake extended it to the *rate* of parallax. Thus, the Radial curvature becomes insensitive to Ω , Ω_t , and \mathbf{c}_{tt} , but it can only be determined relative to a neighboring one. If the neighboring point is viewpoint-independent, then the absolute curvature at an occluding point can be robustly measured.

Implementation of Theory (outline). The system was implemented as a robot arm with a camera, all the imaging parameters being known. The trajectory was approximately linear. Snakes/active contours were initialized by hand on the first hand and tracked. The epipolar geometry is then used for matching (the details of this were not given). Parabola fitting is used to compute temporal derivatives. Reconstruction errors were only given for the radius of transversal curvature.

Summary of Vaillant Faugeras PAMI 1992

This section reviews [150]. We first give the outline of the method, and then explain some important details.

Outline of the method: A static scene is observed by three *static* cameras very close to each other, and the goal is to obtain second-order reconstruction of points around contour generators. The edges are detected using Canny, linked and approximated as polylines. Epipolar geometry is used to match edges. All candidate correspondences are considered at first, and each one corresponds to a putative 3D point. The authors devise criteria for identifying putative contours that are ‘occluding’, based on rough estimate of normal curvature K^r along the visual ray. At occluding contours, they refine depth using rough estimate of K^r . The surface curvatures are re-estimated by computing derivatives, which are obtained by performing surface fitting in the Gauss map orientation space.

The Gauss map is a field of normals on the surface (see Appendix A for a review). First, the authors parametrize the Gauss Map using spherical coordinates:

$$\mathbf{N}(\theta, \phi) = (\cos \theta \cos \phi, \sin \theta \cos \phi, \sin \phi)^\top \quad (2.3.41)$$

One can, then, parametrize the surface by its normals: $\mathbf{\Gamma}(\theta, \phi)$ is the surface point whose normal is $\mathbf{N}(\theta, \phi)$. This is in a neighborhood of all surface points, except at parabolic points, for which an arbitrarily close neighboring point can have the same normal.

The space of measurements: Each image point with attributed tangent gives a tangent plane. The equation of this tangent plane is defined by:

$$\mathbf{N}^w(\theta, \phi) \cdot \mathbf{\Gamma}^w(\theta, \phi) - p(\theta, \phi) = 0, \quad (2.3.42)$$

where p is the distance from the plane to the world origin. The derivatives of p are shown to be related to the derivatives of $\mathbf{\Gamma}^w$ and, ultimately, to the first and second fundamental forms of the surface \mathcal{M} .

The authors devised a numerical method to compute the derivatives of p , thus enabling the computation of the first and second fundamental forms. For each view, each image curve traces a measurement curve in (θ, ϕ, p) parameter space. A surface fitting is then performed in this parameter space. Since the parameters θ, ϕ are periodic, Fourier interpolation is used.

Contour Classification: The problem with applying the aforementioned ideas is that we don't know if a given image curve is the projection of a contour generator. Therefore, we have to find a method for identifying edges that are apparent contours.

The authors devised a rough method to identify potential occluding contours. Only one normal curvature K^r is used, which is along the visual direction. To estimate it, three matching viewing rays are projected onto the radial plane being considered (the radial plane is the plane that contains the viewing ray and the surface normal). The normal curvature at a radial plane is then obtained by constructing an osculating circle tangent to the three rays. The contour classification then boils down to testing the probability of a putative 3D point having non-zero K^r . The uncertainty is approximately modeled from image localization noise, and no camera uncertainties are considered.

If a point is classified as an apparent contour, the depth is refined as the point of contact of the rough osculating circle constructed in the classification stage. Now we can obtain measurements (θ, ϕ, p) as mentioned above and estimate the surface curvatures. The result is a sparse reconstruction with small patches of osculating paraboloids (second-order approximations) around 3D positions.

Summary of Zheng PAMI 1994

This section reviews [163]. We first give the outline of the method without much detail. After that, we explain only the main points in more detail.

Main contributions:

- Contribution 1: Reconstruction of global 3D models. Previous methods only recovered small patches. A sequence of images is acquired from turntable motion. Typically, the radius of rotation is 0.5 meters and full $360 \times 1^\circ$ rotation steps are performed around the object. Only silhouettes (outermost contour generators) are used in the implementation.
- Contribution 2: The method detects where reconstruction from silhouettes is not possible: concave regions, normal discontinuities, *etc.* Spatio-temporal derivatives are computed adaptively by first detecting discontinuities, and then computing the final values without crossing over the discontinuities. The detection of regions where silhouette reconstruction breaks down is useful, and the authors suggest the use of other cues in those cases, such as surface markings, shape from shading, and detection of internal occluding contours.
- Contribution 3: "Scanline parametrization": Instead of epipolar parametrization, Zheng proposes a parametrization where the same raster line (image row) is used to match apparent contours between multiple views. This makes matching very simple in practice, although it complicates much of the mathematics.

The reconstructed surface is represented by triangular patches. These are generated by first reconstructing a point cloud, and then connecting neighboring points with triangles. No curvature is recovered, only orientation. The practical reason the authors give to this is that the computation of the required second-order derivatives is unfeasible. However, it seems that there is a theoretical

reason as well – that the curvature formulas would be too complicated for the proposed parametrization. We recall that, for the epipolar parametrization, the curvature formulas simplify because the coordinate curves are in conjugate directions. However, the directions of the coordinate curves proposed by Zheng are more complicated, even though the matching is simpler. The reason Zheng didn't use pairwise epipolar geometry is to save computation time, and also because he claims its computation is inaccurate for a dense image sequence. In my opinion, even though his parametrization is seemingly practical because there is no need to rectify the images, it may not be the best option for computing surface curvatures.

It is not clear what reconstruction errors are involved. What would be the reconstructed 3D point? What is the advantage over using a standard triangulation? This should be more emphasized in this and other papers. Until now, the only reason for using silhouettes in this method (and many others) is that they are easy to detect under controlled conditions, and that it is easier to reconstruct even when there are no surface markings to track.

The surface can be parametrized by $\Gamma(s, t)$. The “ s -curve” is the curve obtained by varying s and fixing $t = t_0$, given by $\Gamma(s) = \Gamma(s, t_0)$. The t -curve is the one obtained by varying t and keeping $s = s_0$ fixed, $\Gamma(t) = \Gamma(s_0, t)$. As explained in Section 3.2.5, the s -curve is generally the contour generator corresponding to the view at $t = t_0$. The choice for the t -curve is arbitrary. Zheng's paper picks the t -curve as the space curve on \mathcal{M} that projects to the same scanline on different images. At two different instants $t = t_0$, $t = t_0 + dt$, we have that if $\Gamma(s, t)$ projects to the scanline with coordinate $\eta = \eta_0$ of the image at $t = t_0$, then $\Gamma(s, t + dt)$ will project to the scanline η_0 of the image at $t = t + dt$, for $dt \rightarrow 0$. Thus, $\frac{\partial \eta}{\partial t} = 0$ for this parametrization, where $\eta(s, t)$ is the vertical coordinate of the projection of $\Gamma(s, t)$, *i.e.*, $\eta(s, t) = \frac{y(s, t)}{z(s, t)}$ assuming a calibrated camera (so that focal length is normalized to unit).

The measurement space can be visualized as a sequence of stacked silhouettes through time, called the *spatio-temporal volume*. For each fixed η , the intersection of the $t - \xi$ plane with the observed silhouettes is called the “ t image curve”. For each fixed t , the silhouette observed in each image at time t is called the “ t image curve”. The cameras all look into the axis of rotation, so it is projected as a ξ -constant plane in the spatio-temporal volume.

The authors derived a depth reconstruction formula for the scanline parametrization, which we reproduce below using our notation. We start with two basic equations – the occluding contour condition and the normal equation:

$$(\Gamma^w - \mathbf{c}) \cdot \mathbf{N}^w = 0 \quad (2.3.43)$$

$$\mathbf{N}^w = \mathcal{R}^\top \frac{\gamma \times \gamma_s}{\|\gamma \times \gamma_s\|} \quad (2.3.44)$$

which are derived in Section 3.2.5. For convenience, we use the vector $\tilde{\mathbf{N}}^w$ in the same direction of \mathbf{N}^w but without the normalizing factor:

$$(\Gamma^w - \mathbf{c}) \cdot \tilde{\mathbf{N}}^w = 0 \quad (2.3.45)$$

$$\tilde{\mathbf{N}}^w = \mathcal{R}^\top (\gamma \times \gamma_s). \quad (2.3.46)$$

Differentiate the occluding contour equation with respect to s and t :

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \tilde{\mathbf{N}}^w = 0 \quad (2.3.47)$$

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \tilde{\mathbf{N}}_t^w + (\mathbf{\Gamma}_t^w + \mathbf{c}_t) \cdot \tilde{\mathbf{N}}^w = 0 \quad (\text{derivative with respect to } t) \quad (2.3.48)$$

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \tilde{\mathbf{N}}_s^w + \mathbf{\Gamma}_s^w \cdot \tilde{\mathbf{N}}^w = 0 \quad (\text{derivative with respect to } s) \quad (2.3.49)$$

Using $\mathbf{\Gamma}_t^w \cdot \tilde{\mathbf{N}}^w = \mathbf{\Gamma}_s^w \cdot \tilde{\mathbf{N}}^w = 0$, we have:

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \tilde{\mathbf{N}}^w = 0 \quad (2.3.50)$$

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \tilde{\mathbf{N}}_t^w + \mathbf{c}_t \cdot \tilde{\mathbf{N}}^w = 0 \quad (2.3.51)$$

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \tilde{\mathbf{N}}_s^w = 0, \quad (2.3.52)$$

which gives three equations in the unknown $\mathbf{\Gamma}^w = (x^w, y^w, z^w)^\top$. In other words, $\mathbf{\Gamma}^w$ is determined as the intersection of 3 planes, whose normals are $\tilde{\mathbf{N}}^w$, $\tilde{\mathbf{N}}_s^w$, and $\tilde{\mathbf{N}}_t^w$, respectively. We compute $\tilde{\mathbf{N}}_s^w$ and $\tilde{\mathbf{N}}_t^w$ by differentiating Equation (2.3.44):

$$\tilde{\mathbf{N}}_t^w = \mathcal{R}_t^\top (\boldsymbol{\gamma} \times \boldsymbol{\gamma}_s) + \mathcal{R}^\top (\boldsymbol{\gamma}_t \times \boldsymbol{\gamma}_s + \boldsymbol{\gamma} \times \boldsymbol{\gamma}_{st}) \quad (2.3.53)$$

$$\tilde{\mathbf{N}}_s^w = \mathcal{R}^\top (\boldsymbol{\gamma} \times \boldsymbol{\gamma}_{ss}) \quad (2.3.54)$$

We can plug the general formulas for $\tilde{\mathbf{N}}^w$, $\tilde{\mathbf{N}}_t^w$ and $\tilde{\mathbf{N}}_s^w$ into (2.3.50):

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \mathcal{R}^\top (\boldsymbol{\gamma} \times \boldsymbol{\gamma}_s) = 0 \quad (2.3.55)$$

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot [\mathcal{R}_t^\top (\boldsymbol{\gamma} \times \boldsymbol{\gamma}_s) + \mathcal{R}^\top (\boldsymbol{\gamma}_t \times \boldsymbol{\gamma}_s + \boldsymbol{\gamma} \times \boldsymbol{\gamma}_{st})] + \mathbf{c}_t \cdot (\boldsymbol{\gamma} \times \boldsymbol{\gamma}_s) = 0 \quad (2.3.56)$$

$$(\mathbf{\Gamma}^w - \mathbf{c}) \cdot \mathcal{R}^\top (\boldsymbol{\gamma} \times \boldsymbol{\gamma}_{ss}) = 0. \quad (2.3.57)$$

Since Zheng chooses t to be the rotation angle of the turntable, \mathcal{R} has a simple form which can be trivially differentiated. However, to keep the formulas general, we will indicate the derivative of \mathcal{R} by \mathcal{R}_t . Furthermore, Zheng models the apparent contour as a function of η ,¹ *i.e.*, the parameter $s = \eta$, so that

$$\boldsymbol{\gamma}(s, t) = (\xi(s, t), \eta(s, t), 1)^\top = (\xi(s, t), s, 1)^\top. \quad (2.3.58)$$

Thus, $\boldsymbol{\gamma}_s = (\xi_s, 1, 0)$ and $\boldsymbol{\gamma}_t = (\xi_t, 0, 0)$. These identities can be used to get a specific reconstruction formula, which is not reproduced here because it is just an arbitrary specialization of the formulas above.

In summary, the overall strategy for obtaining the reconstruction formula is: (i) differentiate the occluding contour condition with respect to t and s , obtaining two more equations dependent on derivatives of $\tilde{\mathbf{N}}^w$; (ii) Compute these derivatives $\tilde{\mathbf{N}}_t^w$ and $\tilde{\mathbf{N}}_s^w$ from the normal formula and plug them in the equations we had. We will end up with three equations involving $\mathbf{\Gamma}^w$ as three unknowns, all else being observables.

¹It is being assumed that image curves as 2D functions of η . The t image curves are badly sampled when the tangent to the s curves are nearly horizontal, *i.e.*, for high $\|d\eta/ds\|$

Summary of Joshi *et. al.* IJCV 1999

This section reviews [80]. A trinocular stereo rig observes a scene. The relative pose of these 3 cameras is known, and the overall motion of the rig plus 3D structure are recovered by the method. A local paraboloid is used to model the surface around the point of a contour generator. This model can be estimated using the trinocular rig at a single time instant, using a similar technique to [150]. The method breaks down for points near frontiers. The motion parameters are refined by a downhill simplex numerical scheme; given the current motion solution, it minimizes the difference between (i) predicted matches using epipolar geometry, and (ii) paraboloid-induced matches.

Initial solution to camera motion: The initial solution to the motion of the trinocular rig is obtained by tracking apparent contour inflections. As shown by [88], these inflections are projections of parabolic points of the surface \mathcal{M} . It is shown that a constraint can be imposed on the rotation. This constraint involves the known asymptotic direction and the known normal. The initial translation is very coarsely initialized by the following procedure. First, the epipolar plane is roughly estimated from the current estimate of rotation, as if we had orthographic projection. From this, epipolar tangencies can be matched, and the translation is solved-for linearly.

The classification of edges is performed before applying the aforementioned reconstruction approach, and proceeds as follows. First, from two views, reproject into a third; if the predicted position is too far from the expected one, then mark the edge as an apparent contour edge.

Summary of Szeliski Weiss IJCV 1998

This section reviews [143]. The method assumes a linear camera motion path. The epipolar plane is used for the estimation of the radial curvature K^r . The main contribution is the robustness of the method. Tools like Kalman filtering and smoothing are used in order to make optimal use of each measurement.

Summary of Boyer Berger IJCV 1997

This section reviews [16]. The authors propose a more careful numerical scheme for computing the depth at apparent contours. Many methods discretize the depth formula directly, but this is equivalent to treating the underlying contour generator as fixed in 3D. Therefore, there would be *no* advantage of using occluding contours here, as far as precision of depth reconstruction is concerned. A more precise formula is possible that takes advantage of the occluding contours and the information they provide about the curvature of the underlying surface.

2.4 Other Works Using Curves

Summary of Carceroni and Kutulakos CVPR 1999

This section reviews [22]. This paper deals with video sequences of rigid or non-rigid curves observed by two or more cameras over time. Given a calibrated multi-view video of a non-rigid moving curve, the goal of the paper is to find its 3D shape and motion (3D velocity). A special case is a multiple camera rig moving rigidly observing a static curve, where the goal is to estimate the 3D curve and the motion of the camera rig. The paper explores usual multiview constraints at each time instant, together with a spatio-temporal constraint which hasn't been used before.

Similar ideas were later used for local surface patches by the same authors [23]. Material related to those papers also appeared in [21]. According to the authors, previous work either dealt only with monocular sequences of contours [48], or used plain multiview stereo at each time instant, without relating different instants to constrain the problem. Note, however, that this is not strictly true, since [48] does have a section that deals with a stereo rig observing curves, see a review in Section 2.1.2.

An outline of the proposed practical system is as follows:

- Pick a reference time $t = 0$ and a reference viewpoint, and one curve point γ .
- For each curve point, epipolar geometry at $t = 0$ gets corresponding points across frames
- Linear reconstruction of Γ minimizing distance to 3D rays.
- For the next time instant $t = dt$, look around γ in the reference view for candidate curve points, and for each candidate
 - Use epipolar geometry and triangulate Γ at $t = dt$
 - Write a spatio-temporal consistency equation for each view:

$$\begin{cases} \text{unknowns: } \mathbf{\Gamma}^w(t=0), \mathbf{\Gamma}_t^w(t=0) \\ \text{observables: } \gamma, \beta, \mathbf{n}, \mathbf{t}, R^{(i)}, \mathcal{T}^{(i)}, \end{cases} \quad (2.4.1)$$

where super indices represent view number.

For a fixed time instant, it is obvious that the constraint is the usual stereo / epipolar constraint: each 3D point of Γ must project into the observed ones; or, equivalently, the correspondences must be along epipolar lines. How do the authors derive a temporal constraint at different time instants? Intuitively, this would consist in writing spatio-temporal constraints for each viewpoint's video stream, and intersect these equations for all viewpoints. Lets delve one more level of detail and see how the authors do it.

As we described in Section 2.1.2, only the normal component of γ_t is observable, which we denote $\gamma_t^N = \beta \mathbf{n}$. It can be shown (although not done in the paper) that the 3D component \mathbf{N}_V of the 3D velocity $\mathbf{\Gamma}_t^w$ is completely determined by the 2D point $\gamma, \beta \mathbf{n}$, and the camera. The 3D

point $\mathbf{\Gamma}$ can be obtained through stereo at the given time instant. Knowledge of \mathbf{N}_V constrains the 3D point $\tilde{\mathbf{\Gamma}}(dt) = \mathbf{\Gamma}_0 + \mathbf{\Gamma}_t^w(0)dt$ to be on a plane, called the **motion constraint plane**.

More specifically, in 3D, $\mathbf{\Gamma}_t^w$ can be decomposed into

$$\mathbf{\Gamma}_t^w = \mathbf{T}_V + \mathbf{N}_V, \quad (2.4.2)$$

defined for the camera center $\mathbf{c}(t)$ (sub-index is needed if more than 1 camera at time t). The tangential component \mathbf{T}_V is defined to lie on the plane defined by γ and the 3D tangent \mathbf{T} to Γ . The normal component is defined to \mathbf{N}_V lie along $\gamma \times t$.

Observation 1. (no proof in the paper) $\tilde{\mathbf{\Gamma}}(dt)$ is contained in a so-called **motion constraint plane** completely determined by $\mathbf{\Gamma}$, γ , and βn .

Observation 2. If we use multiple cameras at the same time instant, then there are multiple motion constraint planes all intersecting on the same line. This line contains $\tilde{\mathbf{\Gamma}}(dt)$ and is parallel to the curve's tangent \mathbf{T} at $\mathbf{\Gamma}$. It is called the **multiview constraint line**.

As we said, there are two types of constraints to reconstruct 3D structure and motion at a given reference time instant $t = 0$:

1. **stereo constraint:** $\mathbf{\Gamma}(0)$ must project to its images $\gamma_i(0)$ in all views, $i = 1, \dots, n_v$.
2. **motion constraint:** $\mathbf{\Gamma}(0) + \mathbf{\Gamma}_t(0)dt$ must lie on $\mathbf{\Gamma}$'s multiview motion constraint line.

The two constraints are coupled, since the motion constraint line depends on the 3D position $\mathbf{\Gamma}$. In practice: find $\mathbf{\Gamma}, \tilde{\mathbf{\Gamma}}(dt)$ that minimize their least squares distance from appropriately chosen 3D lines that represent the stereo and motion constraints at $\mathbf{\Gamma}$. This is a 3D-error minimizing approach, rather than a maximum likelihood approach that minimizes 2D image-based errors. The advantage of the proposed approach is the solvability through simple linear systems. In light of this, we can rephrase the constraints:

1. **stereo constraint:** Find the 3D point $\mathbf{\Gamma}$ that minimizes the perpendicular distance to 3D rays through its observed projections γ . This is expressed by a linear system in $\mathbf{\Gamma}$.
2. **motion constraint:** This is also expressed as a linear system in the unknowns $\mathbf{\Gamma}$ and $\tilde{\mathbf{\Gamma}}(dt)$, as demonstrated in the appendix of the paper.
3. **stereo + motion constraints:** The motion and stereo constraints are combined into a single linear system, since both share the same unknown $\mathbf{\Gamma}$. The system is over-constrained: 8 constraints on 6 unknowns.

The authors first detect and link edges in each image, and find putative corresponding points in 3 views using the known epipolar geometry. For each putative point correspondence, the paper proposes to solve the combined linear system independently, 3D point and motion hypotheses. The conjectured 3D points are rank-ordered by the residual in the system solution (although image-based would have been better). In summary: they are using spatio-temporal information to improve 3D

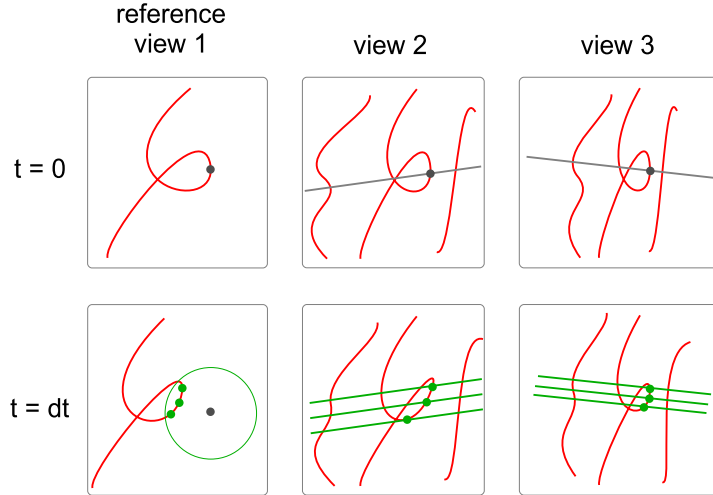


Figure 2.4: Figure illustrating how the stereo and motion constraints are used in [22].

reconstruction *and* disambiguate matches. It could even be used for tracking. (See the algorithm description in the paper).

Stereo matching and disambiguation is done both in fixed time, and across time. One way to interpret the practical implementation of this system is:

1. For $t = 0$, multiview correspondence gives $\mathbf{\Gamma}(0)$ by stereo reconstruction.
2. For $t = dt$, multiview correspondence gives an approximate $\tilde{\mathbf{\Gamma}}(dt)$ by stereo reconstruction.
Pick $\mathbf{\Gamma}(dt)$ minimizing stereo-motion constraints.

Note that the paper does not deal with any ambiguities in stereo for a fixed time instant – it is assumed one can determine the correspondences unambiguously. The only ambiguity arises when searching for the corresponding image point in the next frame.

Experiments on synthetic data with added noise show that the method provides slight improvements on the precision of reconstruction when compared to pure stereo approach. This suggests that a coupled estimation of shape and motion counteracts 3D shape errors due to the localization of individual projected curve points. More specifically, since errors due to localization are generally independent across images, the coupled estimation of shape and motion allows the inclusion of images from multiple time instants in the shape estimation process, providing additional independent constraints to counteract localization errors.

Derivation of the motion constraint plane. The motion constraint plane is just a single temporal equation in the 3 unknowns of $\mathbf{\Gamma}_t$ that has to be satisfied at a view, with coefficients given by measurable quantities (normal motion). As usual, let super-indices denote the view number. Then:

$$\mathbf{\Gamma}^{(i)} = \rho^{(i)} \boldsymbol{\gamma}^{(i)} \quad (2.4.3)$$

$$R^{(i)} \mathbf{\Gamma}^w + \mathcal{T}^{(i)} = \rho^{(i)} \boldsymbol{\gamma}^{(i)}. \quad (2.4.4)$$

Differentiating both sides with respect to time,

$$R^{(i)} \mathbf{\Gamma}_t^w + \mathcal{T}^{(i)} = \rho_t^{(i)} \boldsymbol{\gamma}^{(i)} + \rho^{(i)} \boldsymbol{\gamma}_t^{(i)}. \quad (2.4.5)$$

Taking the dot product with $\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}$,

$$(R^{(i)} \mathbf{\Gamma}_t^w + \mathcal{T}^{(i)})^\top (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}) = (\rho^{(i)} \boldsymbol{\gamma}_t^{(i)})^\top (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}). \quad (2.4.6)$$

Expand $\boldsymbol{\gamma}_t = (\beta \mathbf{n} + \alpha \mathbf{t})$, so that

$$(R^{(i)} \mathbf{\Gamma}_t^w + \mathcal{T}^{(i)})^\top (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}) = (\rho^{(i)} \beta \mathbf{n})^\top (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}). \quad (2.4.7)$$

$$\mathbf{\Gamma}_t^{w\top} R^{(i)\top} (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}) + \mathcal{T}^{(i)\top} (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}) = \rho^{(i)} \beta^{(i)} \mathbf{n}^{(i)\top} (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}) \quad (2.4.8)$$

and, since $\rho^{(i)} = \mathbf{e}_3^{(i)\top} (R^{(i)} \mathbf{\Gamma}^w + \mathcal{T}^{(i)})$, we have

$$\mathbf{\Gamma}_t^{w\top} R^{(i)\top} (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}) + \mathcal{T}^{(i)\top} (\boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)}) = \mathbf{e}_3^{(i)\top} (R^{(i)} \mathbf{\Gamma}^w + \mathcal{T}^{(i)}) \beta^{(i)} \mathbf{n}^{(i)\top} \boldsymbol{\gamma}^{(i)} \times \mathbf{t}^{(i)} \quad (2.4.9)$$

The latter expression is an equation involving the unknowns $\mathbf{\Gamma}^w$ and $\mathbf{\Gamma}_t^w$, whose coefficients are only measurable quantities in the image (including normal velocity). It is called the ‘‘motion constraint equation’’ because, if $\mathbf{\Gamma}^w$ is given, then it is a single linear equation in the 3 coordinates of $\mathbf{\Gamma}_t^w$, which can be interpreted as a plane. Therefore, it seems that three frames would be sufficient to get the full $\mathbf{\Gamma}_t^w$. Note, however, that we know that $\mathbf{\Gamma}_t^w$ is ambiguous, and only its normal component can be unambiguously defined! A more careful inspection of the above equation shows that at each view, $\boldsymbol{\gamma}$ and \mathbf{t} contains the 3D tangent \mathbf{T} . For each view, the component of $\mathbf{\Gamma}_t^w$ along \mathbf{T} is unconstrained. Thus, we are determining only the component of $\mathbf{\Gamma}_t^w$ that lies in the plane having normal \mathbf{T} . Therefore, two views should be enough, as only 2DOF can be determined for $\mathbf{\Gamma}_t^w$. The remaining 1DOF ambiguity is the motion constraint line mentioned in the paper. It cannot be resolved no matter how many views are used.

Summary of Ebrahimnezhad and Ghassemian IVC 2008

The paper [37] proposes a method of reconstructing 3D objects by moving them in front of a calibrated system of two stereo heads perpendicular to each other. First, curves are detected in each image, then matched across frames of the same stereo head using curvature and torsion temporal consistency, generating 3D curves at the current instant. The motion parameters (rotation and translation) are then iteratively estimated such that the 3D curves project to image edges in the next time instant. The rotation and translation then allows to generate virtual cameras, from which the silhouettes are reprojected in 3D and the visual hull of the object is constructed.

The curves are used in order to obtain a photometrically-robust matching and motion estimation. Once the motion is estimated, the curves themselves are not reconstructed for the final model, only the visual hull.

The correspondence of curves between two views is disambiguated using both epipolar geometry and temporal consistency: at each time instant, any two curves can go together, but only the right correspondence can be described by a global 3D rigid motion through time. Since curvature and torsion are rigid motion invariants, they are used in order to compare putative space curves at each time instant – thus a temporal curvature and torsion consistency is enforced. For the curves that match, a bundle adjustment scheme on the rotation matrix and translation vector between the current frame and the next one is estimated, with the reprojection error being defined as smallest distance to image curve points.

The perpendicular stereo configuration is designed in a way that minimizes the uncertainty in the motion estimation. No edge or correspondence information is used across the stereo heads; the motion is estimated in a global coordinate frame, but reprojection is always being measured within the same stereo head.

Chapter 3

Multiview Differential Geometry of Curves

3.1 Introduction

The 3D reconstruction of scenes from images taken from multiple cameras or a moving camera is a fundamental problem in Computer Vision. The reconstruction requires a knowledge of the correspondence of image structures across views, the relative pose of cameras, and their intrinsic calibration parameters. The latter two problems in turn require the knowledge of correspondence for a certain number of structures. The question then is: what sort of image structure can be reliably correlated across images, in the sense that correlated image structures should arise from the same 3D structure? One approach is to consider the correspondence at every point of the image. Another approach is to extract and correlate isolated keypoints (interest points). Yet another approach relies on the extraction and correlation of curvilinear structures in the form of curve fragments or complete silhouettes. These three main approaches can be viewed as correlating 2D, 0D, and 1D structures, respectively, each of which is summarized below.

1) **Keypoint-based methods** extract a set of isolated point features which are expected to be somewhat stable with view variations. These interest points became popular in a paradigm shift away from full segmentation of the image and toward capturing some essential, quasi-invariant representation that can be used in multiview geometry and object recognition. Specifically, points that satisfy certain local conditions in the spatial and scale dimensions, such as Harris corners [67,112] and SIFT [101], *etc.*, are extracted (see review [110]), and attributed with a local description of the image relative to the scale of the interest point (see review of local descriptors [111]). While many of these attributed feature points are not stable with view changes in that they disappear/appear or change in location or description, sufficiently many are stable enough to drive a matching process. The selection of the right subset to match is handled by trying small groups of features through RANSAC [54], which can then be validated by enumerating the number and quality of the inliers in

the remaining features. This set of correspondences is then used to initialize camera models, which are in turn refined through bundle adjustment [1, 69, 125].

A major drawback of using interest points for reconstruction is that they are sparse, resulting not in a mesh but in an unorganized cloud of 3D points where the geometric structure of the underlying curves and surfaces is not explicit, Figure 3.1(a). This is not much of a problem if the focus is on calibrating the viewpoints as in Phototourism [1]. However, in numerous applications such as modeling for architecture, archaeology, entertainment, object recognition, and robotic manipulation, the explicit 3D object geometry is required.¹ A second major drawback of using interest points is that the baseline between views is typically limited, according to some estimates no greater than 30° [113], Figure 3.1(b). In contrast, certain curve features, such as sharp ridges on a building, persist over a much greater range of views. A third major drawback is that the approach relies on an abundance of features so that some of them survive the various variations between views. While this is true in many scenes, as evidenced by the popularity of this approach, in numerous others this is not the case, such as (i) Homogeneous regions, *e.g.*, from man-made objects, corridors, *etc.*, Figure 3.1(c,d); (ii) Multiple moving objects require their own set of features which may not be sufficiently abundant without sufficient texture, Figure 3.1(e); (iii) Non-rigid objects require a rich set of features per roughly non-deforming patch, Figure 3.1(f). In all these cases, however, there is sufficient image curve structure, motivating augmenting the use of interest points with that of image curve structure.

2) Pixel-based multiview stereo methods aim to correlate each image pixel in each view so that a dense and detailed 3D surface can be reconstructed with explicit geometric neighborhood structure in the form of a mesh (see [134] for a review). In volumetric approaches [74, 97], visibility and photometric consistency of the observed image patches are estimated for each voxel of a predefined 3D volume containing the object of interest, and a final surface is then extracted from this volume. Other techniques iteratively evolve an initial surface for optimizing photo-consistency. The initial surface is usually a simple shape containing a predefined working volume around the object, or is obtained using the visual hull as described in the class of curve-based methods below. Surfaces are evolved either as level-sets, as meshes/snakes, or by carving or adding voxels. Some methods combine the volumetric and surface evolution approaches with the use of a silhouette consistency term [74]. Another class of techniques is the image-space methods that compute a set of depth maps where the intensity pattern around each pixel is used for finding dense correspondence across multiple views. These depth maps are then merged into a 3D scene [61, 134].

Pixel-based multiview stereo methods have a number of drawbacks. First, all these methods require a large number of precisely calibrated cameras. Second, they require that the object of interest be within a working 3D volume/bounding box or within multiple near/far planes, and thus cannot usually handle general scenes or acquisition scenarios (*e.g.*, a handheld camera acquiring an

¹Some approaches mesh a 3D point cloud, particularly using Poisson surface reconstruction [87] where normals are computed at each 3D point using patch-based multiview stereo techniques [58]. However, this approach suffers from oversmoothing artifacts, specially in regions where the 3D points are sparse.

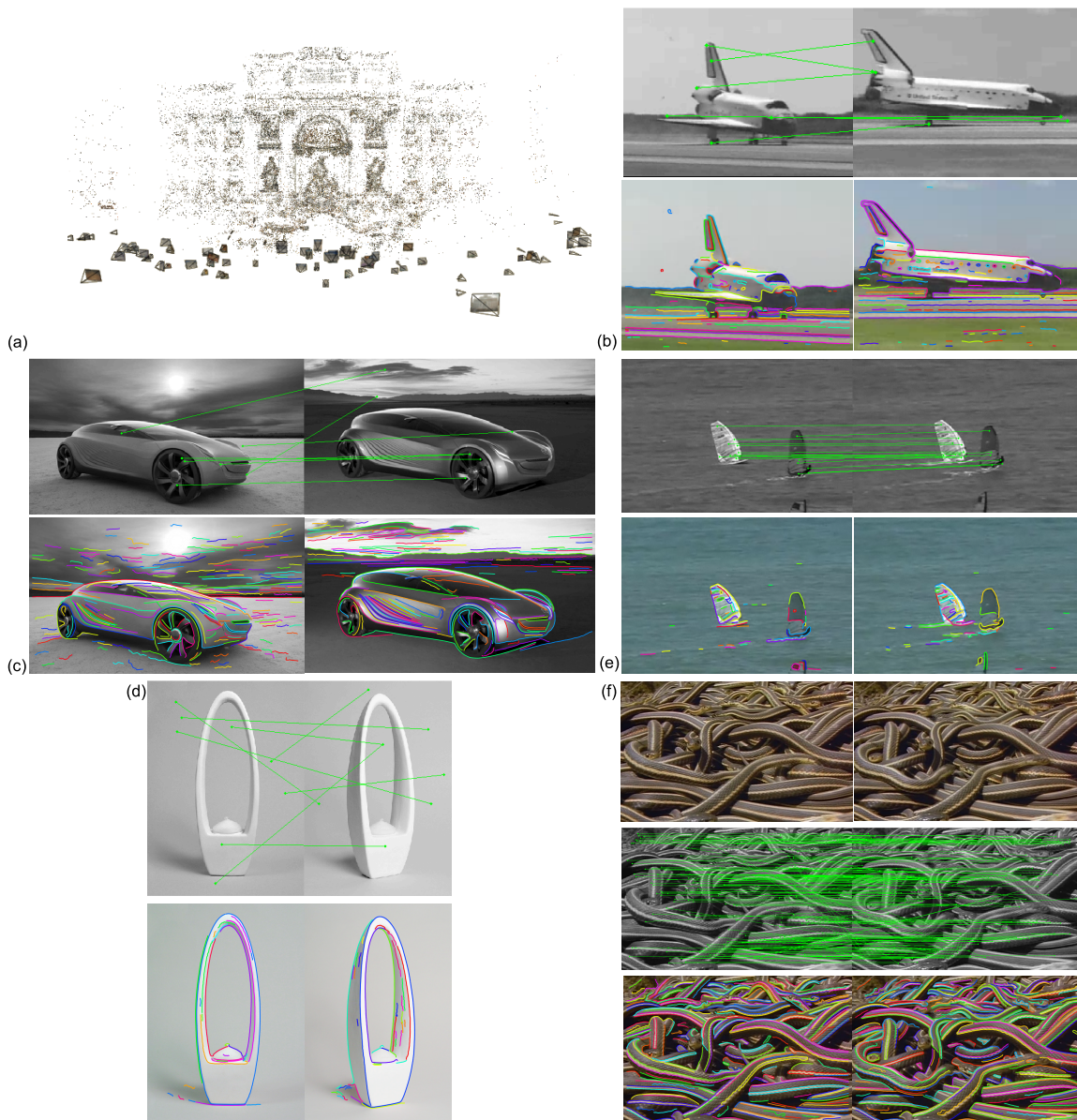


Figure 3.1: (a) Interest point-based approaches give a sparse reconstruction in the form of an unorganized cloud of points; image taken from [1]. (b) Views with wide baseline separation may not have any interest points in common, but they often do share common curve structure. There may not always be sufficient interest points matching across views of homogenous objects such as the car in (c) or the sculpture in (d), but there is sufficient curve structure. (e) each moving object requires its own set of features, but they may not be sufficient without a rich texture surface. (f) Non-rigid structures face the same issue.

outdoor scene). Third, methods making use of silhouette consistency or initializing from the visual hull assume that the image is easily segmentable, requiring that the scene be artificially acquired with a homogeneous background, and typically that the scene has a single object. Fourth, volumetric methods require a large amount of working memory, specially for high-resolution imagery covering a large field of view.

3) Curve-based multiview methods typically require an accurate set of camera models and can be divided into three categories: *(i)* convex hull construction, *(ii)* occluding contour reconstruction, and *(iii)* use of differential geometry in binocular and trinocular stereo. First, when a large of views is available around the object of interest, an initial visual hull is constructed from silhouette curves and then evolved to optimize photometric constraints while constraining the surface projection to the observed silhouettes. The main drawbacks of these methods is that they require controlled acquisition, such as a homogeneous background for automatic silhouette segmentation, require an accurately calibrated system of cameras, impose restrictions on the number and type of objects to be reconstructed, and usually impose a that the object be within predefined 3D boundaries (bounding volume) and that all cameras observe the same object.

Second, the occluding contours extracted from frames of a video are used to reconstruct a local surface model given camera models for each frame. These methods work well for certain applications but require highly controlled acquisition, requiring the image curves to be easy to segment and track in multiple frames, and require very accurate camera models. In addition, since only the silhouettes are used, internal surface variations which may not map to apparent contours in any view will not be captured, *e.g.*, as in certain surface folds of a sculpture. Our approach considers apparent contours as well as other types of contours.

Third, a class of methods use differential geometry in correlating structure across views. Unfortunately, differential geometry does not provide any constraint for matching two points in binocular stereo, as known for tangents and curvatures, and shown here for higher order differential geometry. Heuristics have been used when two views are close to restrict orientation difference [3, 62, 138] or to restrict search based on a locally planar approximation [133]. Differential geometry, however, becomes a powerful tool in trinocular stereo and in multiview stereo in general, as first proposed in [8], who used the constraint that corresponding pairs of points and tangents from two images uniquely determine a point and tangent in a third view to match line segments obtained from edge linking [8]. See also [70, 136, 141]. Robert and Faugeras [131] extended Ayache’s method of transferring points and tangents from two views to a third to include curvature: 3D curvature and normal can be reconstructed from 2D curvatures at two views, which in turn determine the curvature in a third view. The use of curvature results in improved precision and density in the reconstruction, and replaces heuristics such as the ordering constraint [117]. Schmid and Zisserman [133] also derived a formula for transferring curvatures from two views to a third when only the trifocal tensor is available (*i.e.*, when the intrinsic parameters are unknown), using a projective geometry transfer of an osculating circle as a conic.

It should be noted that more recently the differential geometry of curves in two views was

used for reconstruction by imposing an additional constraint that the 3D curves of interest have small torsion [94]. Specifically, the compatibility of two hypothetical pairs of point correspondences with tangent and curvature measurements is related to the torsion of the 3D reconstructed curve. A relaxation labeling process is then used to discover an optimal 3D reconstruction minimizing torsion.

The main drawback of this third class of methods is that they require very precise calibration, and the binocular stereo methods have biased heuristics.

Curves have also been used in the calibration stage using the concept of epipolar tangencies: if an epipolar line is tangent to a curve in one image, its corresponding epipolar line must also be tangent to the corresponding curve. This constraint is used to capture the epipolar geometry (or relative pose if the intrinsic parameters are known). All previous work in this category is restricted to closed curves, however, which can be either images of stationary curves (reflectance curves, sharp ridges, *etc.*, that do not change with view) or of silhouettes.

In summary, the very popular methods based on interest points are powerful in calibration and sparse reconstruction in numerous applications, but are limited in providing explicit 3D geometry, are limited in the baseline between views, and are not always applicable, *e.g.*, if there is not a sufficient number of interest points. Pixel-based methods and curve-based methods using either visual hull or occluding contours require highly controlled acquisition and/or very accurate camera models. What would be desirable is a generic, generally applicable framework, *e.g.*, a handheld video acquiring images around the object of interest or a set of cameras simultaneously monitoring an object, where the image curve structure can be used to calibrate and reconstruct a 3D wireframe on which a surface can be tautly stretched like a tent on a metallic scaffold.

The use of image curve fragments as the basic structure to be correlated across images and as a basic path to general scene reconstruction is not without challenges, however. The process of linking edges into curve fragments is fraught with ambiguities so that there is abundant instability of curve fragments with view variation. Also, even if it is known that two curve fragments correspond, there remains an intra-curve correspondence ambiguity. Despite these drawbacks image curve fragments are attractive because they have good localization, they have greater invariance than interest points to changes in illumination, are stable over a greater range of baselines, and are much denser than interest points. Furthermore, stationary curves such as reflectance or ridge curves provide boundary conditions for surface reconstruction, while occluding contour variation across views directly leads to surface reconstruction [27]. The notion that image curves contain much of the image information is supported by a recent study [29] which found that generally subjects are able to judge 3D surface normals of an object depicted by a line-drawing almost as accurately as for objects depicted by a shaded image.

The clear advantage of using curve structures in images motivates an intermediate representation between isolated points and long/closed curve fragments to address the above drawbacks, namely the use of short portions of curve segments, as fully captured by the differential geometry attributes of a point or a curve. This thesis develops the theoretical foundations required in using the differential geometry for image curve structure as a complementary alternative to using

interest points. Specifically, calibration and reconstruction using differential geometry include: (i) How does the differential geometry of a space curve map to the differential geometry of the image curve it projects to? (ii) How can the differential geometry of a space curve be reconstructed from the differential geometry of two corresponding image curves, each from a different view? (iii) How does the differential geometry of an image curve evolve under differential camera motion, whether it arises from an occluding contour or a non-occluding contour? (iv) How does the differential geometry of a surface map to the differential flow of image points and what does this imply in the vicinity of or at image curves? Clear answers to these questions under a unifying framework would undoubtedly pave the road to the use of short image curve segments in multiview problems. Some of this work has already been done as pointed to earlier, but some has not been done, and some have been erroneously derived, *e.g.*, the equation relating normal contour velocity of non-occluding contours under camera motion [120, 122], which is shown here to have a missing term, and which is extended here to handle occluding contours. This theoretical chapter has been the required foundation of work already reported on the reconstruction of a 3D curve sketch from image fragments in numerous views [47] and is the basis of calibration based on differential geometry of curves which is not published yet [45].

A part of this work appeared in a conference [46]. In this thesis we extend these results to account for differentiable motion, occluding and nonrigid contours, and intrinsic parameters. In future work, we plan to develop a differential scene model by relating the differential geometry of surfaces, differential models of camera motion, and even differential models of illumination to differential observations in the image as organized into *curve fragments* and *interior points* of objects bounded by these curve fragments.

This chapter is organized along the line of the four questions imposed above. Section 3.2 reviews and establishes notation for relevant differential geometry of 2D and 3D curves, camera projection, camera motion, and discusses an important distinction between *stationary* and *non-stationary* 3D contours as projected onto image curves. Section 3.3 derives formulas for obtaining the differential geometry of image curves, tangent, curvature, and curvature derivative from the differential geometry of the space curves they arise from, *i.e.*, tangent and normal, curvature, torsion, and curvature derivatives. Section 3.4 derives the differential geometry of a space curve at a point from the differential geometry at two corresponding image curve points by showing the key result that the ratio of parametrization speeds in the two curves is an intrinsic quantity. It is also notable that each order of differential geometry (first-order is tangent, second-order is curvature, third-order is curvature derivative and torsion) requires only differential geometry of that order or less, *e.g.*, a reconstruction of curvature does not require curvature derivatives of the image curves. The key new result in this section is the reconstruction of torsion and curvature derivative, given corresponding differential geometry in two views.

Section 3.5 relates the differential geometry of a space curve under projection to a camera under differential motion to the image curve differential geometry and differential motion. Results are provided concerning the image velocities and accelerations with respect to time, for points

on different types of curves. In order to distinguish apparent contours from stationary contours, it is necessary to use second-order time derivative. We also study the spatial variation of the image velocity field along curves, providing results which can be useful for exploiting neighborhood consistency of velocity fields along curves. The main new theorem of the section corrects a missing term in the results of [120, 122] and generalizes the equation to also represent occluding contours in addition to stationary contours such as reflectance curves.

Section 3.6 relates the differential geometry of a surface observed from a camera with differential motion to the observation of the spatial structure of the resulting image velocity map (optical flow).

This chapter integrates the above results under the umbrella of one integrated formulation, completes missing relationships, and corrects erroneous previous results. As a generalized framework, this chapter is expected to act as a reference material for research relating local properties of general curves and surfaces to those of cameras and images.

3.2 Notation and Formulation

3.2.1 Differential Geometry of Curves

A 3D space curve Γ is a mapping $S \mapsto \Gamma^w(S)$ from \mathbb{R} to \mathbb{R}^3 , where S is an arbitrary parameter, \tilde{S} is the arc-length parameter, and the superscript w denotes the world coordinates. The local Frenet frame of Γ in world coordinates is defined by tangent \mathbf{T}^w , normal \mathbf{N}^w , binormal \mathbf{B}^w ; G is speed of parametrization, curvature K , and torsion τ . Similarly, a 2D curve γ is a mapping $s \mapsto \gamma(s)$ from \mathbb{R} to \mathbb{R}^2 , where s is an arbitrary parameter, \tilde{s} is arc-length, $g = \|\gamma'\|$ is speed of parametrization, $\mathbf{t} = \gamma'/g$ is tangent, $\mathbf{n} = \mathbf{t}^\perp$ is normal, κ is curvature defined by $g\kappa\mathbf{n} = \mathbf{t}'$, and κ' is curvature derivative. By classical differential geometry [35], we have:

$$\left\{ \begin{array}{l} G = \|\Gamma^{w'}\| \\ \mathbf{T}^w = \frac{\Gamma^{w'}}{G} \\ K = \frac{\|\mathbf{T}^{w'}\|}{G} \end{array} \right. \quad \mathbf{N}^w = \frac{\mathbf{T}^{w'}}{\|\mathbf{T}^{w'}\|} \quad \mathbf{B}^w = \mathbf{T}^w \times \mathbf{N}^w \quad \begin{array}{l} \dot{K} = \frac{K'}{G} \\ \tau = \frac{-\mathbf{B}^{w'} \cdot \mathbf{N}^w}{G} \end{array}, \quad \begin{bmatrix} \mathbf{T}^{w'} \\ \mathbf{N}^{w'} \\ \mathbf{B}^{w'} \end{bmatrix} = G \begin{bmatrix} 0 & K & 0 \\ -K & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \mathbf{T}^w \\ \mathbf{N}^w \\ \mathbf{B}^w \end{bmatrix}. \quad (3.2.1)$$

where throughout the paper prime “'” and dot “.” denote differentiation with respect to the arbitrary spatial parameter (S or s) and to arc-length (\tilde{S} or \tilde{s}), respectively. The matrix equations on the right are the Frenet equations. Note that both the curvature derivatives \dot{K} and $\dot{\kappa} = \kappa'/g$ are intrinsic quantities.

3.2.2 Perspective Projection

The projection of a 3D Space curve Γ into a 2D image curve γ is illustrated by Figure 3.2(a), where the world coordinate system is centered at O with basis vectors $\{\mathbf{e}_1^w, \mathbf{e}_2^w, \mathbf{e}_3^w\}$. The *camera coordinate system* is centered at $\mathbf{c} = [c^x, c^y, c^z]^\top$ with basis vectors $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. When describing coordinates

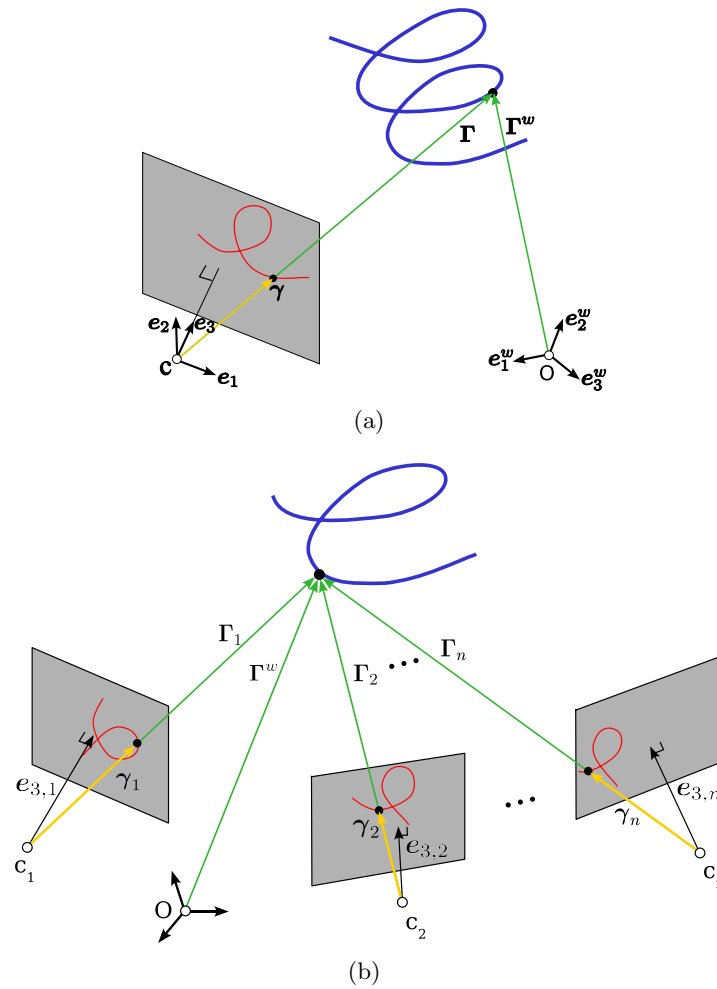


Figure 3.2: (a) The perspective projection of a space curve. (b) The projection of a space curve in n views.

in the camera coordinate system we drop the w superscript, *e.g.*, $\mathbf{\Gamma}$ versus $\mathbf{\Gamma}^w$, which are related by:

$$\mathbf{\Gamma} = \mathcal{R}(\mathbf{\Gamma}^w - \mathbf{c}) = \mathcal{R}\mathbf{\Gamma}^w + \mathcal{T}, \quad (3.2.2)$$

where $\mathcal{T} = -\mathcal{R}\mathbf{c}$ denotes the world coordinate origin in the camera coordinate system.

The projection of a 3D point $\mathbf{\Gamma} = [x, y, z]^\top$ onto the image plane at $z = 1$ is the point $\boldsymbol{\gamma} = [\xi, \eta, 1]^\top$ related by

$$\mathbf{\Gamma} = \rho\boldsymbol{\gamma} \quad \text{or} \quad [x, y, z]^\top = [\rho\xi, \rho\eta, \rho]^\top, \quad (3.2.3)$$

where we say that $\boldsymbol{\gamma}$ is in normalized image coordinates (focal point is normalized to 1), and the depth is $\rho = z = \mathbf{e}_3^\top \mathbf{\Gamma}$ from the third equation. Observe that image points are treated as 3D points with $z = 1$. Thus, we can write

$$\boldsymbol{\gamma} = \frac{\mathbf{\Gamma}}{\mathbf{e}_3^\top \mathbf{\Gamma}}. \quad (3.2.4)$$

We note that $\mathbf{e}_3^\top \boldsymbol{\gamma}^{(i)} = 0$ and $\mathbf{e}_3^\top \mathbf{\Gamma}^{(i)} = \rho^{(i)}$, where $\boldsymbol{\gamma}^{(i)}$ is the i^{th} derivative of $\boldsymbol{\gamma}$ with respect to an arbitrary parameter, for any positive integer i . Specifically,

$$\rho = \mathbf{e}_3^\top \mathbf{\Gamma}, \quad \rho' = G\mathbf{e}_3^\top \mathbf{T}, \quad \rho'' = G'\mathbf{e}_3^\top \mathbf{T} + G^2 K\mathbf{e}_3^\top \mathbf{N}. \quad (3.2.5)$$

It is interesting to note that at near/far points of the curve, *i.e.*, $\rho' = 0$, $\mathbf{e}_3^\top \mathbf{T} = 0$.

In practice, normalized image coordinates $\boldsymbol{\gamma} = [\xi, \eta, 1]^\top$ are described in terms of image pixel coordinates $\boldsymbol{\gamma}_{im} = [x_{im}, y_{im}, 1]^\top$ through the intrinsic parameter matrix \mathcal{K}_{im} :

$$\boldsymbol{\gamma}_{im} = \mathcal{K}_{im}\boldsymbol{\gamma}, \quad \mathcal{K}_{im} = \begin{bmatrix} \alpha_\xi & \sigma & \xi_o \\ 0 & \alpha_\eta & \eta_o \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.2.6)$$

where as usual ξ_o and η_o are the principal points, σ is skew, and α_ξ and α_η are given by the focal length divided by the width and height of a pixel in world units, respectively.

3.2.3 Discrete and Continuous Sets of Views

Two scenarios are considered. The first scenario consists of a *discrete set of views* where a set of n pinhole cameras observe a scene as shown in Figure 3.2(b), with the last subscript in the symbols identifying the camera, *e.g.*, $\boldsymbol{\gamma}_i$ denotes an image point in the i^{th} camera, and $\mathbf{e}_{3,i}$ denotes \mathbf{e}_3 in the i^{th} view. The second scenario consists of a *continuous set of views* from a continuously moving camera observing a space curve which may itself be moving, $\mathbf{\Gamma}^w(S, t) = [x^w(S, t), y^w(S, t), z^w(S, t)]^\top$, where S is the parameter along the curve and t is time, described in the camera coordinate system associated with time t as $\mathbf{\Gamma}(S, t) = [x(S, t), y(S, t), z(S, t)]^\top$, Figure 3.3. For simplicity, we often omit the parameters S or t . Let the camera position over time (*camera orbit*) be described by the space curve $\mathbf{c}(t)$ and the camera orientation by a rotation matrix $\mathcal{R}(t)$. For simplicity, and without loss of generality, we take the camera coordinate system at $t = 0$ to be the world coordinate system,

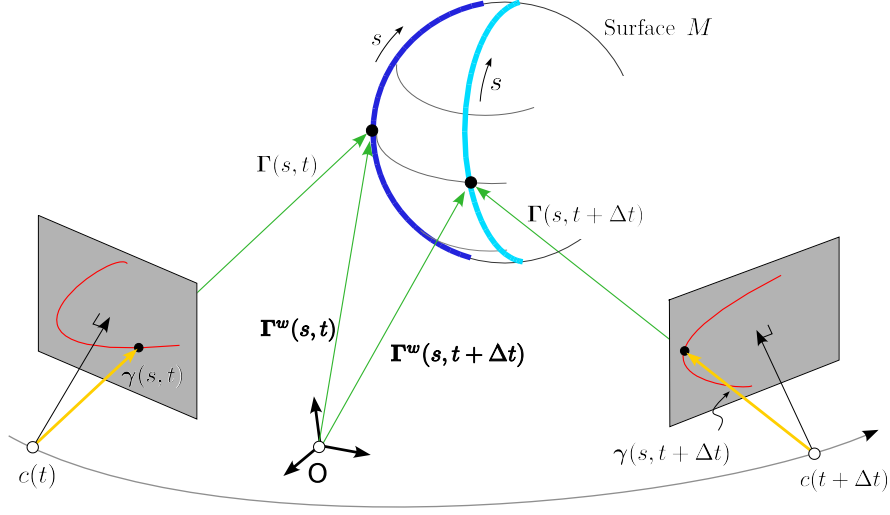


Figure 3.3: Multiview formulation of continuous camera motion and a possibly moving contour.

i.e., $\mathbf{c}(0) = 0$, $\mathcal{T}(0) = 0$, and $\mathcal{R}(0) = I$, where I is the identity matrix. Also, a stationary point can be modeled in this notation by making $\mathbf{\Gamma}^w(t) = \mathbf{\Gamma}^w(0) = \mathbf{\Gamma}_0$.

A differential camera motion model using time derivatives of $\mathcal{R}(t)$ and $\mathcal{T}(t)$ can be used to relate frames in a small time interval. Since $\mathcal{R}\mathcal{R}^\top = I$,

$$\frac{d\mathcal{R}}{dt}\mathcal{R}^\top + \mathcal{R}\frac{d\mathcal{R}^\top}{dt} = 0, \quad (3.2.7)$$

which implies that $\Omega_\times \doteq \frac{d\mathcal{R}}{dt}\mathcal{R}^\top$ is a skew-symmetric matrix, explicitly written as

$$\Omega_\times = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix}, \quad (3.2.8)$$

so that $\frac{d\mathcal{R}}{dt} = \Omega_\times \mathcal{R}$. Denote $\Omega = [\Omega_x, \Omega_y, \Omega_z]^\top$ as a vector form characterization of Ω_\times . Similarly, the second-derivative of $\mathcal{R}(t)$ is represented by only three additional numbers $\frac{d\Omega_\times}{dt}$:

$$\frac{d^2\mathcal{R}}{dt^2} = \frac{d\Omega_\times}{dt}\mathcal{R} + \Omega_\times \frac{d\mathcal{R}}{dt} = \frac{d\Omega_\times}{dt}\mathcal{R} + \Omega_\times^2 \mathcal{R}. \quad (3.2.9)$$

Thus, a second-order Taylor approximation of the camera rotation matrix using $\mathcal{R}(0) = I$ is

$$\mathcal{R}(t) \approx I + \Omega_\times t + \frac{1}{2} \left[\frac{d\Omega_\times}{dt}(0) + \Omega_\times^2(0) \right] t^2. \quad (3.2.10)$$

Similarly, the camera translation can be described by a differential model

$$\mathbf{V}(t) \doteq \frac{d\mathcal{T}}{dt}(t) = -\Omega_\times(t)\mathcal{R}(t)\mathbf{c}(t) - \mathcal{R}(t)\frac{d\mathbf{c}}{dt}, \quad \mathbf{V}(0) = -\frac{d\mathbf{c}}{dt}, \quad (3.2.11)$$

and

$$\frac{d\mathbf{V}}{dt}(t) = \frac{d^2\mathcal{T}}{dt^2}(t) = -\frac{d^2\mathcal{R}}{dt^2}(t)\mathbf{c}(t) - 2\frac{d\mathcal{R}}{dt}(t)\frac{d\mathbf{c}}{dt}(t) - \mathcal{R}(t)\frac{d^2\mathbf{c}}{dt^2}(t), \quad (3.2.12)$$

Symbol	Description	Symbol	Description
Γ^w	3D point in the world coordinate system	\mathbf{t}	Image curve tangent $\mathbf{t} = \boldsymbol{\gamma}'/g$
Γ	3D point in the camera coord. syst. $\Gamma = \mathcal{R}\Gamma^w + \mathcal{T}$	\mathbf{n}	Image curve normal $\mathbf{n} = \mathbf{t}^\perp$
\mathcal{R}	Rotation matrix: world to camera coordinates	κ	Curvature of the image curve $g\kappa\mathbf{n} = \mathbf{t}'$
\mathcal{T}	Translation vector: world to camera coord. $\mathcal{T} = -\mathcal{R}\mathbf{c}$	S, \tilde{S}	Space curve arbitrary parameter & arclength, resp.
\mathbf{c}	The camera center	G	Space curve speed of parametrization $G = \ \Gamma'\ $
Ω_\times	$\frac{d\mathcal{R}}{dt} = \mathcal{R}_\times$	\mathbf{T}, \mathbf{T}^w	Space curve tangent camera & world coord., resp.
Ω	Vector form of the 3 entries of Ω_\times	\mathbf{N}, \mathbf{N}^w	Space curve normal: camera & world coord., resp.
\mathbf{V}	$\mathbf{V} = \frac{d\mathcal{T}}{dt} = \Omega_\times\mathcal{T} - \mathcal{R}\mathbf{c}_t$, also $\mathbf{V} = [V_x, V_y, V_z]^\top$	\mathbf{B}, \mathbf{B}^w	Space curve binormal: camera & world coord., resp.
ρ	Depth of image point $\Gamma = \rho\boldsymbol{\gamma}$	$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$	Basis vectors of the camera coordinate system
$\boldsymbol{\gamma}$	2D point in normalized image coordinates	$\mathbf{e}_1^w, \mathbf{e}_2^w, \mathbf{e}_3^w$	Basis vectors of the world coordinate system
$\boldsymbol{\gamma}_{im}$	2D point in pixel image coordinates	\cdot	Diff. with resp. to S or s
s, \tilde{s}	Image curve arbitrary parameter & arclength, resp.	\cdot	Diff. with resp. to arclength \tilde{S} or \tilde{s}
g	Image curve speed of parametrization $g = \ \boldsymbol{\gamma}'\ $	θ	The angle $\angle(\mathbf{T}, \boldsymbol{\gamma})$
(u, v)	Image velocities $\boldsymbol{\gamma}_t = [u, v, 0]^\top$		

Table 3.1:

which at $t = 0$ gives $\frac{d\mathbf{V}}{dt}(0) = -2\Omega_\times(0)\frac{d\mathbf{c}}{dt}(0) - \frac{d^2\mathbf{c}}{dt^2}(0)$.

The choice of whether to adopt the Taylor approximation of $\mathbf{c}(t)$ or $\mathcal{T}(t)$ as primary is entirely dependent in which domain the higher derivatives are expective to diminish, giving

$$\mathcal{T}(t) = \mathbf{V}(0)t + \frac{1}{2}\mathbf{V}_t(0)t^2 + O(t^3), \quad \mathbf{c}(t) = -\mathbf{V}(0)t + \frac{1}{2}\left[-\frac{d^2\mathbf{c}}{dt^2}(0) + 2\Omega_\times(0)\mathbf{V}(0)\right]t^2. \quad (3.2.13)$$

3.2.4 Relating World and Camera-Centric Derivatives.

Proposition 3.2.1. *The velocity of a 3D point $\Gamma(t)$ in camera coordinates, $\Gamma_t(t)$, is related to its velocity in the world coordinates Γ_t^w by:*

$$\left\{ \begin{array}{l} \Gamma_t = \Omega_\times\Gamma + \mathcal{R}\Gamma_t^w - \mathcal{R}\mathbf{c}_t = \Omega_\times\mathcal{R}\Gamma^w + \mathcal{R}\Gamma_t^w + \mathbf{V}, \end{array} \right. \quad (3.2.14)$$

$$\left\{ \begin{array}{l} \Gamma_t = \Omega_\times\Gamma - \mathcal{R}\mathbf{c}_t = \Omega_\times\mathcal{R}\Gamma_0 + \mathbf{V}, \quad \text{for a fixed point, } \Gamma^w = \Gamma_0. \end{array} \right. \quad (3.2.15)$$

Proof. Differentiating Equation 3.2.2 with respect to time,

$$\Gamma_t = \mathcal{R}_t\Gamma^w + \mathcal{R}\Gamma_t^w + \mathcal{T}_t \quad (3.2.16)$$

$$= \Omega_\times\mathcal{R}\Gamma^w + \mathcal{R}\Gamma_t^w + \mathbf{V}, \quad (3.2.17)$$

$$= \Omega_\times(\Gamma - \mathcal{T}) + \mathcal{R}\Gamma_t^w + \mathbf{V} \quad (3.2.18)$$

$$= \Omega_\times\Gamma + \mathcal{R}\Gamma_t^w + \mathbf{V} - \Omega_\times\mathcal{T}. \quad (3.2.19)$$

The result follows from using $\mathcal{T} = -\mathcal{R}\mathbf{c}$ as below

$$\mathbf{V} = \mathcal{T}_t = -\mathcal{R}_t\mathbf{c} - \mathcal{R}\mathbf{c}_t = -\Omega_\times\mathcal{R}\mathbf{c} - \mathcal{R}\mathbf{c}_t = \Omega_\times\mathcal{T} - \mathcal{R}\mathbf{c}_t. \quad (3.2.20)$$

■

3.2.5 Stationary and Non-Stationary Contours

It is important to differentiate between image contours arising from a space curve that is changing at most with a rigid transform (stationary contours), *e.g.*, reflectance contours and sharp ridges, and image curves arising from deforming space curves (non-stationary contours), *e.g.*, occluding contours, for which the *contour generators* project to *apparent contours*. Stationary contours are characterized by $\mathbf{\Gamma}_t^w = 0$ while for occluding contours the viewing direction $\mathbf{\Gamma}(S, t)$ is tangent to the surface \mathcal{M} with surface normal \mathbf{N} ($\mathbf{N}^w = \mathcal{R}^\top \mathbf{N}$)

$$\mathbf{\Gamma}^\top \mathbf{N} = 0, \quad \text{or} \quad (\mathbf{\Gamma}^w - \mathbf{c})^\top \mathbf{N}^w = 0. \quad (3.2.21)$$

For the image curve $\gamma(s, t)$ arising from the occluding contour, Figure 3.3, the normal \mathbf{N} to an occluding contour [27] is $\mathbf{N} = \frac{\gamma \times \dot{\gamma}}{\|\gamma \times \dot{\gamma}\|}$.

The correlation of the parametrization S of Γ at time t to that of nearby times is captured by $\mathbf{\Gamma}_t^w(S, t)$, which is orthogonal to \mathbf{N} , but is otherwise arbitrary as a one dimensional choice. It is common to require that $\mathbf{\Gamma}_t^w(S, t)$ lay on the (infinitesimal) epipolar plane, spanned by $\mathbf{\Gamma}^w(S, t)$, $\mathbf{c}(t)$, and $\mathbf{c}_t(t)$, referred to as the *epipolar parametrization* [27], requiring

$$\mathbf{\Gamma}_t^w \times (\mathbf{\Gamma}^w - \mathbf{c}) = 0, \quad \text{or} \quad \mathbf{\Gamma}_t^w = \lambda(\mathbf{\Gamma}^w - \mathbf{c}) \text{ for some } \lambda. \quad (3.2.22)$$

3.3 Projecting Differential Geometry Onto a Single View

This section relates the intrinsic differential-geometric attributes of the space curve and those of its perspective image curves. Specifically, the derivatives $\mathbf{\Gamma}'$, $\mathbf{\Gamma}''$, and $\mathbf{\Gamma}'''$ are first expressed in terms of the differential geometry of Γ , namely $\{\mathbf{T}, \mathbf{N}, \mathbf{B}, K, \dot{K}, \tau\}$, and second, they are expressed in terms of the differential geometry of γ , namely $\{\mathbf{t}, \mathbf{n}, \kappa, \dot{\kappa}\}$ using $\mathbf{\Gamma} = \rho\gamma$. Note that \dot{K} and $\dot{\kappa}$ are both intrinsic quantities. In equating these two expressions we relate $\{\mathbf{T}, \mathbf{N}, \mathbf{B}, K, \dot{K}, \tau\}$ to $\{\mathbf{t}, \mathbf{n}, \kappa, \dot{\kappa}\}$. Our purpose is to eliminate the dependence on the parametrizations (g, G), and depth ρ , *i.e.*, final expressions do not contain these unknowns nor their derivatives (g, g', g''), (G, G', G''), or unknown depth and its derivatives ($\rho, \rho', \rho'', \rho'''$).

Proposition 3.3.1. $\{\mathbf{T}, \mathbf{N}, \mathbf{B}, K, \dot{K}, \tau, G, G', G''\}$ are related to $\{\gamma, \mathbf{t}, \mathbf{n}, \kappa, \dot{\kappa}, g, g', g'', \rho, \rho', \rho'', \rho'''\}$ by

$$\begin{cases} G\mathbf{T} = \rho'\gamma + \rho g\mathbf{t} & (3.3.1) \end{cases}$$

$$\begin{cases} G'\mathbf{T} + G^2 K\mathbf{N} = \rho''\gamma + (2\rho'g + \rho g')\mathbf{t} + \rho g^2 \kappa \mathbf{n} & (3.3.2) \end{cases}$$

$$\begin{cases} (G'' - G^3 K^2)\mathbf{T} + (3GG'K + G^3 \dot{K})\mathbf{N} + G^3 K\tau \mathbf{B} = \\ \rho''' \gamma + [3\rho''g + 3\rho'g' + \rho(g'' - g^3 \kappa^2)]\mathbf{t} + [3\rho'g^2 \kappa + \rho(3gg'\kappa + g^3 \dot{\kappa})]\mathbf{n}. & (3.3.3) \end{cases}$$

Proof. First, writing $\mathbf{\Gamma}'$, $\mathbf{\Gamma}''$, and $\mathbf{\Gamma}'''$ in the Frenet frame of $\mathbf{\Gamma}$ as

$$\begin{cases} \mathbf{\Gamma}' = G\mathbf{T} & (3.3.4) \end{cases}$$

$$\begin{cases} \mathbf{\Gamma}'' = G'\mathbf{T} + G^2 K\mathbf{N} & (3.3.5) \end{cases}$$

$$\begin{cases} \mathbf{\Gamma}''' = (G'' - G^3 K^2)\mathbf{T} + (3GG'K + G^2 \dot{K})\mathbf{N} + G^3 K\tau \mathbf{B}. & (3.3.6) \end{cases}$$

Note that when expressed with respect to the arc-length of Γ , i.e., $G \equiv 1$, simple expressions result:

$$\begin{cases} \dot{\Gamma} = T & (3.3.7) \\ \ddot{\Gamma} = KN & (3.3.8) \\ \dddot{\Gamma} = -K^2T + \dot{K}N + K\tau B. & (3.3.9) \end{cases}$$

Second, differentiating $\Gamma = \rho\gamma$ gives:

$$\begin{cases} \Gamma' = \rho'\gamma + \rho\gamma' & (3.3.10) \\ \Gamma'' = \rho''\gamma + 2\rho'\gamma' + \rho\gamma'' & (3.3.11) \\ \Gamma''' = \rho'''\gamma + 3\rho''\gamma' + 3\rho'\gamma'' + \rho\gamma'''. & (3.3.12) \end{cases}$$

This can be rewritten using expressions for the derivatives of γ , which are

$$\begin{cases} \gamma' = g\mathbf{t} & (3.3.13) \\ \gamma'' = g'\mathbf{t} + g^2\kappa\mathbf{n} & (3.3.14) \\ \gamma''' = (g'' - g^3\kappa^2)\mathbf{t} + (3gg'\kappa + g^2\kappa')\mathbf{n}. & (3.3.15) \end{cases}$$

Thus, Γ' , Γ'' , and Γ''' can be written in terms of γ , \mathbf{t} , \mathbf{n} , κ , $\dot{\kappa}$, g , g' , g'' , ρ , ρ' , ρ'' , ρ''' as

$$\begin{cases} \Gamma' = \rho'\gamma + \rho g\mathbf{t} & (3.3.16) \\ \Gamma'' = \rho''\gamma + (2\rho'g + \rho g')\mathbf{t} + \rho g^2\kappa\mathbf{n} & (3.3.17) \\ \Gamma''' = \rho'''\gamma + [3\rho''g + 3\rho'g' + \rho(g'' - g^3\kappa^2)]\mathbf{t} \\ \quad + [3\rho'g^2\kappa + \rho(3gg'\kappa + g^3\dot{\kappa})]\mathbf{n}. & (3.3.18) \end{cases}$$

Equating (3.3.4-3.3.6) and (3.3.16-3.3.18) proves the proposition. \blacksquare

Corollary 3.3.2. *Using the arc-length \tilde{S} of the space curve as the common parameter, i.e., when $G \equiv 1$, we have:*

$$\begin{cases} T = \rho'\gamma + \rho g\mathbf{t} & (3.3.19) \\ KN = \rho''\gamma + (2\rho'g + \rho g')\mathbf{t} + \rho g^2\kappa\mathbf{n} & (3.3.20) \\ -K^2T + \dot{K}N + K\tau B = \rho'''\gamma + [3\rho''g + 3\rho'g' + \rho(g'' - g^3\kappa^2)]\mathbf{t} \\ \quad + [3\rho'g^2\kappa + \rho(3gg'\kappa + g^3\dot{\kappa})]\mathbf{n}. & (3.3.21) \end{cases}$$

First-Order Differential Geometry. We are now in a position to derive the first-order differential attributes of the image curve (g, \mathbf{t}) from that of the space curve (G, \mathbf{T}) . Note from (3.3.1) or (3.3.19) that \mathbf{T} lies on the plane spanned by \mathbf{t} and γ , i.e., \mathbf{T} is a linear combination of these vectors. An exact relationship is expressed below.

Theorem 1. *Given the tangent \mathbf{T} at Γ when \mathbf{T} is not aligned with γ , then the corresponding tangent \mathbf{t} and normal \mathbf{n} at γ are determined by:*

$$\mathbf{t} = \frac{\mathbf{T} - (e_3^\top \mathbf{T})\gamma}{\|\mathbf{T} - (e_3^\top \mathbf{T})\gamma\|}, \quad \mathbf{n} = \mathbf{t}^\perp \doteq \mathbf{t} \times \mathbf{e}_3. \quad (3.3.22)$$

Proof. First recall that $\rho' = \mathbf{e}_3^\top \mathbf{\Gamma}' = G \mathbf{e}_3^\top \mathbf{T}$. Using this in Equation 3.3.1 gives:

$$\mathbf{t} = \frac{1}{\rho g} [G\mathbf{T} - \rho' \boldsymbol{\gamma}] = \frac{G}{\rho g} [\mathbf{T} - \mathbf{e}_3^\top \mathbf{T} \boldsymbol{\gamma}] \quad (3.3.23)$$

and the result follows by normalizing. The formula for the normal comes from the fact that it lies in the image plane, therefore being orthogonal to both \mathbf{t} and \mathbf{e}_3 . ■

Observe that the depth scale factor ρ is not needed to find \mathbf{t} from \mathbf{T} . Moreover, when $\boldsymbol{\gamma}$ and \mathbf{T} are aligned for a point on γ , Equation (3.3.23) still holds, but implies that $g = 0$ and \mathbf{t} is undefined, *i.e.*, that the image curve will have stationary points and possibly corners or cusps. Stationary points are in principle not detectable from the trace of $\boldsymbol{\gamma}$ alone, but by the assumption of general position these do not concern us.

A crucial quantity in relating differential geometry along the space curve to that of the projected image curve is the *ratio of speeds of parametrizations* $\frac{g}{G}(s)$. The following theorem derives the key result that this quantity is *intrinsic* in that it does not depend on either $g(s)$ or $G(s)$ at each arbitrary s , thus allowing a relationship between the differential geometry of the space and image curves.

Theorem 3.3.3. *The ratio of speeds of the projected 2D curve g and of the 3D curve G at corresponding points is an intrinsic quantity:*

$$\frac{g}{G} = \frac{\|\mathbf{T} - (\mathbf{e}_3^\top \mathbf{T}) \boldsymbol{\gamma}\|}{\mathbf{e}_3^\top \mathbf{\Gamma}} \quad \text{or} \quad g = \frac{\|G\mathbf{T} - \rho' \boldsymbol{\gamma}\|}{\rho}. \quad (3.3.24)$$

i.e., it does not depend on the parametrization of $\mathbf{\Gamma}$ or of $\boldsymbol{\gamma}$.

Proof. Follows from a dot product of Equation 3.3.23 with \mathbf{t} and dividing by $\rho = \mathbf{e}_3^\top \mathbf{\Gamma}$. ■

Second-Order Differential Geometry. The curvature of an image curve can be derived from the curvature of the space curve, as shown by the next theorem.

Theorem 2. *The curvature κ of a projected image curve is given by:*

$$\kappa = \left[\frac{\mathbf{N} - (\mathbf{e}_3^\top \mathbf{N}) \boldsymbol{\gamma}}{\rho g^2} \cdot \mathbf{n} \right] K, \quad \text{when } G \equiv 1, \quad \text{or} \quad \kappa = G \left[\frac{\mathbf{N}^\top (\boldsymbol{\gamma} \times \mathbf{t})}{\rho g^2 \mathbf{n}^\top (\boldsymbol{\gamma} \times \mathbf{t})} \right] K, \quad (3.3.25)$$

where g is given by Equation 3.3.24 and $\rho = \mathbf{e}_3^\top \mathbf{\Gamma}$. The tangential acceleration of a projected curve with respect to the arc length of the space curve is given by:

$$\frac{dg}{d\tilde{S}} = \frac{[\mathbf{N} - (\mathbf{e}_3^\top \mathbf{N}) \boldsymbol{\gamma}]^\top \mathbf{t} K}{\rho} - 2g \frac{\mathbf{e}_3^\top \mathbf{T}}{\rho}, \quad \text{or} \quad \frac{dg}{d\tilde{S}} = \frac{K \mathbf{N}^\top (\boldsymbol{\gamma} \times \mathbf{n})}{\rho \mathbf{t}^\top (\boldsymbol{\gamma} \times \mathbf{n})} - 2g \frac{\mathbf{e}_3^\top \mathbf{T}}{\rho}. \quad (3.3.26)$$

Proof. Using Equation 3.2.5 in Equation 3.3.20 leads to

$$GKN = (G' \mathbf{e}_3^\top \mathbf{T} + G^2 K \mathbf{e}_3^\top \mathbf{N}) \boldsymbol{\gamma} + 2(G' \mathbf{e}_3^\top \mathbf{T}) g \mathbf{t} + \rho g' \mathbf{t} + \rho g^2 \kappa \mathbf{n}. \quad (3.3.27)$$

First, in the case of $G \equiv 1$ curvature κ can be isolated by taking the dot product of the last equation with \mathbf{n} which gives the curvature projection formula (3.3.25). Alternatively, taking the dot product with $\boldsymbol{\gamma} \times \mathbf{t}$ gives the alternative formula. Second, the term g' can be isolated by taking the dot product with \mathbf{t} or with $\boldsymbol{\gamma} \times \mathbf{n}$. ■

Note that formulas for the projection of 3D tangent and curvatures onto 2D tangent and geodesic curvature appear in [28] and [27, pp. 73–75], but an actual image curvature was not determined there. That the curvature of the space curve is related to the curvature of the projected curve was derived in previous work [94, 131], but our proof is much simpler and more direct. Moreover, our proof methodology generalizes to relating higher order derivatives such as curvature derivative and torsion, as shown below.

Theorem 3. *The curvature derivative of a projected image curve γ is derived from the local third-order differential geometry of the space curve as follows:*

$$\dot{\kappa} = \frac{[\dot{K}\mathbf{N} + K\tau\mathbf{B}]^\top(\gamma \times \mathbf{t})}{\rho g^3 \mathbf{n}^\top(\gamma \times \mathbf{t})} - 3\kappa \left(\frac{\mathbf{e}_3^\top \mathbf{T}}{\rho g} + \frac{g'}{g^2} \right). \quad (3.3.28)$$

Proof. Taking the scalar product of Equation 3.3.21 with $\gamma \times \mathbf{t}$, and using $\mathbf{T}^\top(\gamma \times \mathbf{t}) = 0$,

$$[\dot{K}\mathbf{N} + K\tau\mathbf{B}]^\top(\gamma \times \mathbf{t}) = [3\rho'g^2\kappa + \rho(3gg'\kappa + g^3\dot{\kappa})]\mathbf{n}^\top(\gamma \times \mathbf{t}), \quad (3.3.29)$$

which using $\rho' = \mathbf{e}_3^\top \mathbf{T}$ gives

$$3\mathbf{e}_3^\top \mathbf{T}g^2\kappa + \rho(3gg'\kappa + g^3\dot{\kappa}) = \frac{[\dot{K}\mathbf{N} + K\tau\mathbf{B}]^\top(\gamma \times \mathbf{t})}{\mathbf{n}^\top(\gamma \times \mathbf{t})}. \quad (3.3.30)$$

Isolating $\dot{\kappa}$ gives the desired result. Since both g and g' are available from Equations 3.3.24 and 3.3.26, the theorem follows. \blacksquare

3.3.1 Intrinsic Parameters and Differential Geometry

This section derives the relationship between the intrinsic differential geometry $\{\mathbf{t}, \mathbf{n}, \kappa, \dot{\kappa}\}$ of the curve in normalized image coordinates to those in image pixel coordinates, $\{\mathbf{t}_{im}, \mathbf{n}_{im}, \kappa_{im}, \dot{\kappa}_{im}\}$. Using the intrinsic parameter matrix \mathcal{K}_{im} relating $\gamma_{im} = \mathcal{K}_{im}\gamma$, by the linear Equation 3.2.6.

Theorem 4. *The Intrinsic quantities $\{\mathbf{t}, \mathbf{n}, \kappa, \dot{\kappa}\}$ and $\{\mathbf{t}_{im}, \mathbf{n}_{im}, \kappa_{im}, \dot{\kappa}_{im}\}$ under linear transformation $\gamma_{im} = \mathcal{K}_{im}\gamma$ are related by*

$$\begin{cases} g_{im} = \|\mathcal{K}_{im}\mathbf{t}\|, & \mathbf{t}_{im} = \frac{\mathcal{K}_{im}\mathbf{t}}{\|\mathcal{K}_{im}\mathbf{t}\|}, & \mathbf{n}_{im} = \mathbf{t}_{im} \times \mathbf{e}_3 & (3.3.31) \\ g'_{im} = \frac{\kappa\mathbf{t}^\top \mathcal{K}_{im}^\top \mathcal{K}_{im} \mathbf{n}}{g_{im}}, & \kappa_{im} = \frac{\mathbf{n}_{im}^\top \mathcal{K}_{im} \kappa \mathbf{n}}{g_{im}^2}, & & (3.3.32) \\ \dot{\kappa}_{im} = \frac{1}{g_{im}^3} \mathbf{n}_{im}^\top \mathcal{K}_{im} (-\kappa^2 \mathbf{t} + \dot{\kappa} \mathbf{n}) - \frac{3g'_{im} \kappa_{im}}{g_{im}^2}. & & & (3.3.33) \end{cases}$$

where the speed g_{im} is relative to unit speed at γ .

Proof. Differentiating (3.2.6) with respect to the arclength \tilde{s} of γ , and using Equation 3.3.16, $\gamma'_{im} = \mathcal{K}_{im}\dot{\gamma}$ gives

$$g_{im}\mathbf{t}_{im} = \mathcal{K}_{im}\mathbf{t}. \quad (3.3.34)$$

Differentiating (3.2.6) a second time with respect to \tilde{s} , and using Equation 3.3.14

$$g'_{im} \mathbf{t}_{im} + g''_{im} \kappa_{im} \mathbf{n}_{im} = \mathcal{K}_{im} \kappa \mathbf{n}. \quad (3.3.35)$$

Taking the dot product with \mathbf{t}_{im} gives the formula for g'_{im} , and taking the dot product with \mathbf{n}_{im} gives the formula for κ_{im} . Differentiating (3.2.6) a third time with respect to \tilde{s} , and using Equation 3.3.15, we have:

$$(g''_{im} - g'''_{im} \kappa_{im}^2) \mathbf{t}_{im} + (3g_{im} g'_{im} \kappa_{im} + g'''_{im} \dot{\kappa}_{im}) \mathbf{n}_{im} = \mathcal{K}_{im} (-\kappa^2 \mathbf{t} + \dot{\kappa} \mathbf{n}). \quad (3.3.36)$$

Taking the dot product with \mathbf{n}_{im} :

$$3g_{im} g'_{im} \kappa_{im} + g'''_{im} \dot{\kappa}_{im} = \mathbf{n}_{im}^\top \mathcal{K}_{im} (-\kappa^2 \mathbf{t} + \dot{\kappa} \mathbf{n}), \quad (3.3.37)$$

and isolating κ_{im} , the last result follows. \blacksquare

The above theorem can also be used in its inverse form from γ_{im} to γ by substituting \mathcal{K}_{im} for \mathcal{K}_{im}^{-1} , and trivially exchanging the sub-indices. Moreover, the theorem is generally valid for relating differential geometry under any linear transformation in place of \mathcal{K}_{im} .

3.4 Reconstructing Differential Geometry from Multiple Views

In the previous section we derived the differential geometry of a projected curve from a space curve. In this section, we derive the differential geometry of a space curve Γ from that of its projected image curves in multiple views, namely γ_i for camera i , $i = 1, \dots, N$. In order to simplify the equations, in this section *all vectors are written in the common world coordinate basis*, including γ_i . Denote $\mathbf{\Gamma}_i := \mathbf{\Gamma}^w - \mathbf{c}_i$, namely $\mathbf{\Gamma}_i$ represents the vector from the i^{th} camera center to the 3D point $\mathbf{\Gamma}^w$ in the world coordinate system.

The reconstruction of a point on the space curve Γ from two corresponding image curve points γ_1 and γ_2 can be obtained by equating the two expressions for $\mathbf{\Gamma}^w$, given by Equation 3.2.3

$$\begin{cases} \mathbf{\Gamma}^w - \mathbf{c}_1 = \rho_1 \gamma_1 \\ \mathbf{\Gamma}^w - \mathbf{c}_2 = \rho_2 \gamma_2 \end{cases} \implies \rho_1 \gamma_1 - \rho_2 \gamma_2 = \mathbf{c}_2 - \mathbf{c}_1. \quad (3.4.1)$$

Taking the dot product with γ_1 , γ_2 , and $\gamma_1 \times \gamma_2$ gives

$$\begin{cases} \rho_1 \gamma_1 \cdot \gamma_1 - \rho_2 \gamma_1 \cdot \gamma_2 = (\mathbf{c}_2 - \mathbf{c}_1) \cdot \gamma_1 \\ \rho_1 \gamma_1 \cdot \gamma_2 - \rho_2 \gamma_2 \cdot \gamma_2 = (\mathbf{c}_2 - \mathbf{c}_1) \cdot \gamma_2 \\ 0 = (\mathbf{c}_2 - \mathbf{c}_1) \cdot (\gamma_1 \times \gamma_2) \end{cases} \quad (3.4.2)$$

which gives

$$\begin{cases} \rho_1 = \frac{(\mathbf{c}_2 - \mathbf{c}_1) \cdot \gamma_1 (\gamma_2 \cdot \gamma_2) - (\mathbf{c}_2 - \mathbf{c}_1) \cdot \gamma_2 (\gamma_1 \cdot \gamma_2)}{(\gamma_1 \cdot \gamma_1)(\gamma_2 \cdot \gamma_2) - (\gamma_1 \cdot \gamma_2)^2} \\ \rho_2 = \frac{(\mathbf{c}_2 - \mathbf{c}_1) \cdot \gamma_1 (\gamma_1 \cdot \gamma_2) - (\mathbf{c}_2 - \mathbf{c}_1) \cdot \gamma_2 (\gamma_1 \cdot \gamma_1)}{(\gamma_1 \cdot \gamma_1)(\gamma_2 \cdot \gamma_2) - (\gamma_1 \cdot \gamma_2)^2}, \end{cases} \quad (3.4.3)$$

provided that $(\mathbf{c}_2 - \mathbf{c}_1) \cdot (\boldsymbol{\gamma}_1 \times \boldsymbol{\gamma}_2) = 0$. This is precisely the well-known fact that this system of three equations in two unknowns ρ_1 and ρ_2 can only be solved if the lines $\mathbf{c}_1\boldsymbol{\gamma}_1$ and $\mathbf{c}_2\boldsymbol{\gamma}_2$ intersect.

The crucial factor in relating the differential geometry of image curves in distinct views is the relationship between parametrization in each view:

Proposition 3.4.1. *The ratio of parametrization speeds in two views of a space curve at corresponding points is given by:*

$$\frac{g_1}{g_2} = \frac{\rho_2}{\rho_1} \frac{\|\mathbf{T} - (\mathbf{e}_{3,1}^\top \mathbf{T})\boldsymbol{\gamma}_1\|}{\|\mathbf{T} - (\mathbf{e}_{3,2}^\top \mathbf{T})\boldsymbol{\gamma}_2\|}. \quad (3.4.4)$$

Proof. Follows by dividing expressions for $\frac{g_1}{G}$ and $\frac{g_2}{G}$ from Equation 3.3.24. ■

Next, note from Equation 3.3.1 that the unit vector \mathbf{T} can be written as

$$\mathbf{T} = \frac{\rho'}{G}\boldsymbol{\gamma} + \rho\frac{g}{G}\mathbf{t}. \quad (3.4.5)$$

Since \mathbf{T} is a unit vector, it can be written as

$$\mathbf{T} = \cos\theta \frac{\boldsymbol{\gamma}}{\|\boldsymbol{\gamma}\|} + \sin\theta \mathbf{t}, \quad \text{where} \quad \cos\theta = \frac{\rho'}{G}\|\boldsymbol{\gamma}\|, \quad \sin\theta = \rho\frac{g}{G}. \quad (3.4.6)$$

Note that $\rho > 0$ implies that $\sin\theta \geq 0$ or $\theta \in [0, \pi)$. Thus the reconstruction of \mathbf{T} from \mathbf{t} requires the discovery of the additional parameter θ which can be provided from tangents at two corresponding points:

Theorem 5. *Two tangent vectors at a corresponding pair of points, namely \mathbf{t}_1 at $\boldsymbol{\gamma}_1$ and \mathbf{t}_2 at $\boldsymbol{\gamma}_2$, reconstruct the corresponding space tangent \mathbf{T} at $\boldsymbol{\Gamma}$*

$$\mathbf{T} = \cos\theta_1 \frac{\boldsymbol{\gamma}_1}{\|\boldsymbol{\gamma}_1\|} + \sin\theta_1 \mathbf{t}_1 = \cos\theta_2 \frac{\boldsymbol{\gamma}_2}{\|\boldsymbol{\gamma}_2\|} + \sin\theta_2 \mathbf{t}_2, \quad (3.4.7)$$

and

$$\begin{aligned} \rho_1 g_1 = \sin\theta_1 G, & \quad \rho'_1 \|\boldsymbol{\gamma}_1\| = \cos\theta_1 G, & \quad \frac{\rho'_1}{\rho_1 g_1} = -\frac{\mathbf{t}_1 \cdot (\boldsymbol{\gamma}_2 \times \mathbf{t}_2)}{\boldsymbol{\gamma}_1 \cdot (\boldsymbol{\gamma}_2 \times \mathbf{t}_2)} \\ \rho_2 g_2 = \sin\theta_2 G, & \quad \rho'_2 \|\boldsymbol{\gamma}_2\| = \cos\theta_2 G, & \quad \frac{\rho'_2}{\rho_2 g_2} = -\frac{\mathbf{t}_2 \cdot (\boldsymbol{\gamma}_1 \times \mathbf{t}_1)}{\boldsymbol{\gamma}_2 \cdot (\boldsymbol{\gamma}_1 \times \mathbf{t}_1)}, \end{aligned} \quad (3.4.8)$$

where

$$\begin{aligned} \tan\theta_1 &= -\frac{1}{\|\boldsymbol{\gamma}_1\|} \frac{\boldsymbol{\gamma}_1 \cdot (\boldsymbol{\gamma}_2 \times \mathbf{t}_2)}{\mathbf{t}_1 \cdot (\boldsymbol{\gamma}_2 \times \mathbf{t}_2)}, & \quad \theta_1 \in [0, \pi) \\ \tan\theta_2 &= -\frac{1}{\|\boldsymbol{\gamma}_2\|} \frac{\boldsymbol{\gamma}_2 \cdot (\boldsymbol{\gamma}_1 \times \mathbf{t}_1)}{\mathbf{t}_2 \cdot (\boldsymbol{\gamma}_1 \times \mathbf{t}_1)}, & \quad \theta_2 \in [0, \pi). \end{aligned} \quad (3.4.9)$$

Proof. Equating the two expressions for \mathbf{T} from Equation 3.4.6, one for each view, gives Equation 3.4.7. Solving for θ_1 by taking the dot product with $\boldsymbol{\gamma}_2 \times \mathbf{t}_2$ gives

$$\cos\theta_1 \frac{\boldsymbol{\gamma}_1}{\|\boldsymbol{\gamma}_1\|} \cdot (\boldsymbol{\gamma}_2 \times \mathbf{t}_2) + \sin\theta_1 \mathbf{t}_1 \cdot (\boldsymbol{\gamma}_2 \times \mathbf{t}_2) = 0, \quad (3.4.10)$$

which leads to Equation 3.4.9 and similarly for θ_2 . Equation 3.4.4 follows from Equation 3.4.6. ■

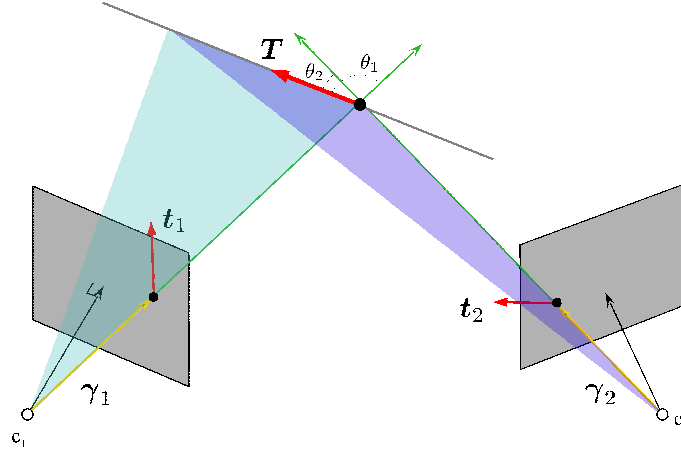


Figure 3.4: 3D Tangent reconstruction from two views as the intersection of two planes.

Remark: Since \mathbf{T} is orthogonal to both $\gamma_1 \times \mathbf{t}_1$, and $\gamma_2 \times \mathbf{t}_2$ we have

$$\varepsilon \mathbf{T} = \frac{(\mathbf{t}_1 \times \gamma_1) \times (\mathbf{t}_2 \times \gamma_2)}{\|(\mathbf{t}_1 \times \gamma_1) \times (\mathbf{t}_2 \times \gamma_2)\|} \quad \varepsilon = \pm 1, \quad (3.4.11)$$

where ε is determined from

$$\begin{cases} \varepsilon [\mathbf{T} - (\mathbf{T} \cdot \mathbf{e}_{3,1}) \gamma_1] \cdot \mathbf{t}_1 > 0 \\ \varepsilon [\mathbf{T} - (\mathbf{T} \cdot \mathbf{e}_{3,2}) \gamma_2] \cdot \mathbf{t}_2 > 0. \end{cases} \quad (3.4.12)$$

Remark: This theorem implies that *any* two tangents at corresponding points can be consistent with at least one space tangent. Furthermore, the discovery of \mathbf{T} does not require the *a priori* solution of ρ_1 or ρ_2 .

Theorem 6. *The normal vector \mathbf{N} and curvature K of a point on a space curve Γ with point-tangent-curvature at projections in two views $(\gamma_1, \mathbf{t}_1, \kappa_1)$ and $(\gamma_2, \mathbf{t}_2, \kappa_2)$ are given by solving the following system in the vector $\mathbf{N}K$:*

$$\begin{cases} G(\gamma_1 \times \mathbf{t}_1)^\top \mathbf{N}K = \mathbf{n}_1^\top (\gamma_1 \times \mathbf{t}_1) \rho_1 g_1^2 \kappa_1 \\ G(\gamma_2 \times \mathbf{t}_2)^\top \mathbf{N}K = \mathbf{n}_2^\top (\gamma_2 \times \mathbf{t}_2) \rho_2 g_2^2 \kappa_2 \\ \mathbf{T}^\top \mathbf{N}K = 0, \end{cases} \quad (3.4.13)$$

where \mathbf{T} is given by Equation 3.4.5, ρ_1 and ρ_2 by Equation 3.4.3, and g_1 and g_2 by Equation 3.3.24.

Proof. Taking the dot product of (3.3.27) with $\gamma \times \mathbf{t}$, for each view, we arrive at the first two equations. The third equation imposes the solution $\mathbf{N}K$ to be normal to \mathbf{T} . ■

The next theorem shows how to reconstruct torsion and curvature derivative from a third order representation of two image curves.

Theorem 7. *The torsion and curvature derivative at a point of a space curve can be obtained from up to third order differential geometry $\{\tau, \kappa, \dot{\kappa}\}$ at a pair of corresponding points by solving for \mathbf{V} in the following system:*

$$\begin{cases} (\boldsymbol{\gamma}_1 \times \mathbf{t}_1)^\top \mathbf{V} = [3g_1^2 \kappa_1 \mathbf{e}_{3,1}^\top \mathbf{T} + \rho_1(3g_1 g_1' \kappa_1 + g_1^3 \dot{\kappa}_1)] \mathbf{n}_1^\top (\boldsymbol{\gamma}_1 \times \mathbf{t}_1) \\ (\boldsymbol{\gamma}_2 \times \mathbf{t}_2)^\top \mathbf{V} = [3g_2^2 \kappa_2 \mathbf{e}_{3,2}^\top \mathbf{T} + \rho_2(3g_2 g_2' \kappa_2 + g_2^3 \dot{\kappa}_2)] \mathbf{n}_2^\top (\boldsymbol{\gamma}_2 \times \mathbf{t}_2) \\ \mathbf{T}^\top \mathbf{V} = 0, \end{cases} \quad (3.4.14)$$

with \mathbf{T} , \mathbf{N} , \mathbf{B} , K , g_1 , g_2 , g_1' , g_2' , ρ_1 , and ρ_2 determined from previous derivations, and by solving for the torsion τ and curvature derivative \dot{K} from $\mathbf{V} = \dot{K}\mathbf{N} + K\tau\mathbf{B}$, i.e.,

$$\begin{cases} \tau = \frac{\mathbf{V}^\top \mathbf{B}}{K} & (3.4.15) \\ \dot{K} = \mathbf{V}^\top \mathbf{N}. & (3.4.16) \end{cases}$$

Proof. Apply Equation (3.3.29) for two views, and let $\mathbf{V} := \dot{K}\mathbf{N} + K\tau\mathbf{B}$ to get the first two equations of (3.4.14). The last equations constrains \mathbf{V} to be orthogonal to \mathbf{T} . ■

3.5 Projecting Differential Geometry Under Differential Motion

The goal of this section is to relate *differential observations* in a series of images from a continuous video sequence to the *differential geometry* of the space curve. As this relationship is governed by the *differential motion* of the camera and its intrinsic parameters, we also aim to recover calibration, scene geometry, and camera motion/pose from these observations. We explore how differential scene properties are projected onto differential image properties for points and curves, and expect future work to apply this to surfaces.

Differential models of camera motion observing a *rigid scene* were studied in [4, 12, 18, 19, 36, 76, 83, 100, 104, 106, 147, 151, 152, 154, 155]. These papers studied how the first and second-order motion of the image of fixed points relate to a differential camera motion model. They also envisioned recovering local 3D surface properties from the local behavior of the velocities of projected surface points in an image neighborhood. Differential models for *nonrigid curves* observed in a monocular video sequence were studied in [48, 120, 122], where it was established that multiple simultaneous video sequences would be needed. This led to a practical work in the reconstruction of nonrigid curves from multiview video [21, 22], exploiting temporal consistency within each video frame, as well as consistency across the two video sequences. Differential models of occluding contours were studied mainly in [24, 27], relating the deformation of apparent contours under differential camera motion to scene properties such as occluding contours, and a differential-geometric model of the underlying 3D surface.

3.5.1 Differential Relations for a Point

Theorem 3.5.1. (*Moving 3D point*) Let $\Gamma^w(t)$ be a moving point in space, projected onto a moving camera as $\gamma(t)$ with depth $\rho(t)$. Let the differential velocity and rotation of the camera be \mathbf{V} and Ω , respectively, and let \mathbf{V}_t and Ω_t represent their derivative with respect to time t , respectively. Then, the depth gradient and second derivative at $t = 0$ are

$$\begin{cases} \rho_t = \rho \mathbf{e}_3^\top (\Omega_\times \gamma + \frac{1}{\rho} \Gamma_t^w + \frac{1}{\rho} \mathbf{V}) \\ \rho_{tt} = \rho \mathbf{e}_3^\top (\Omega_\times^2 + [\Omega_t]_\times) \gamma + 2\mathbf{e}_3^\top \Omega_\times \Gamma_t^w + \mathbf{e}_3^\top \Gamma_{tt}^w + \mathbf{e}_3^\top \mathbf{V}_t, \end{cases} \quad \text{at } t = 0 \quad (3.5.1)$$

$$(3.5.2)$$

and the velocity and acceleration of the projected point at $t = 0$ are given by

$$\begin{cases} \gamma_t = (\Omega_\times \gamma - (\mathbf{e}_3^\top \Omega_\times \gamma) \gamma) + \frac{1}{\rho} (\Gamma_t^w - \mathbf{e}_3^\top \Gamma_t^w \gamma) + \frac{1}{\rho} (\mathbf{V} - \mathbf{e}_3^\top \mathbf{V} \gamma) \\ \gamma_{tt} = (\Omega_\times^2 + [\Omega_t]_\times) \gamma + \frac{2}{\rho} \Omega_\times \Gamma_t^w + \frac{1}{\rho} \Gamma_{tt}^w + \frac{1}{\rho} \mathbf{V}_t - \frac{2\rho_t}{\rho} \gamma_t - \frac{\rho_{tt}}{\rho} \gamma, \end{cases} \quad \text{at } t = 0 \quad (3.5.3)$$

$$(3.5.4)$$

which can be simplified as

$$\begin{aligned} \gamma_{tt} &= (\Omega_\times^2 + [\Omega_t]_\times) \gamma + \frac{2}{\rho} \Omega_\times \Gamma_t^w + \frac{1}{\rho} \Gamma_{tt}^w + \frac{1}{\rho} \mathbf{V}_t \\ &\quad - 2\mathbf{e}_3^\top \left(\Omega_\times \gamma + \frac{\mathbf{V}}{\rho} + \frac{\Gamma_t^w}{\rho} \right) \left(\frac{\mathbf{V}}{\rho} - \frac{\mathbf{e}_3^\top \mathbf{V}}{\rho} \gamma + \Omega_\times \gamma - (\mathbf{e}_3^\top \Omega_\times \gamma) \gamma + \frac{1}{\rho} \Gamma_t^w - \frac{\mathbf{e}_3^\top \Gamma_t^w}{\rho} \gamma \right) \\ &\quad - \mathbf{e}_3^\top \left((\Omega_\times^2 + [\Omega_t]_\times) \gamma + \frac{\mathbf{V}_t}{\rho} + 2\Omega_\times \frac{\Gamma_t^w}{\rho} + \frac{\Gamma_{tt}^w}{\rho} \right) \gamma. \end{aligned} \quad (3.5.5)$$

Proof. The image velocity γ_t is dependent on the velocity Γ_t of the 3D structure in camera coordinates, which arises from both the motion of Γ^w and from the moving camera. Differentiating $\Gamma = \mathcal{R}\Gamma^w + \mathcal{I}$, we get

$$\Gamma_t = \mathcal{R}_t \Gamma^w + \mathcal{R} \Gamma_t^w + \mathcal{I}_t = \mathcal{R} \Omega_\times \Gamma^w + \mathcal{R} \Gamma_t^w + \mathbf{V}. \quad (3.5.6)$$

Differentiating $\Gamma = \rho\gamma$ we get

$$\Gamma_t = \rho\gamma_t + \rho_t\gamma. \quad (3.5.7)$$

Equating these two expressions leads to

$$\rho\gamma_t + \rho_t\gamma = \mathcal{R} \Omega_\times \Gamma^w + \mathcal{R} \Gamma_t^w + \mathbf{V} \quad \text{for arbitrary } t. \quad (3.5.8)$$

At $t = 0$ we have $\Gamma^w = \Gamma = \rho\gamma$, leading to

$$\rho\gamma_t + \rho_t\gamma = \rho \Omega_\times \gamma + \Gamma_t^w + \mathbf{V} \quad \text{for } t = 0. \quad (3.5.9)$$

The depth gradient ρ_t is then isolated by taking the dot product of both sides of Equation 3.5.9 with \mathbf{e}_3 , observing that $\mathbf{e}_3^\top \gamma = 1$ and $\mathbf{e}_3^\top \gamma_t = 0$, resulting in Equation 3.5.1. The expression for ρ_t is then substituted into Equation 3.5.9 from which γ_t can be isolated in the form of Equation 3.5.3.

The second order expressions γ_{tt} and ρ_{tt} require another time derivative of Equation 3.5.8:

$$\rho\gamma_{tt} + 2\rho_t\gamma_t + \rho_{tt}\gamma = (\mathcal{R} \Omega_\times^2 + \mathcal{R} [\Omega_t]_\times) \Gamma^w + \mathcal{R} \Omega_\times \Gamma_t^w + \mathcal{R}_t \Gamma_t^w + \mathcal{R} \Gamma_{tt}^w + \mathbf{V}_t. \quad (3.5.10)$$

Setting $t = 0$ we have

$$\rho\gamma_{tt} + 2\rho_t\gamma_t + \rho_{tt}\gamma = (\Omega_\times^2 + [\Omega_t]_\times)\rho\gamma + 2\Omega_\times\Gamma_t^w + \Gamma_{tt}^w + \mathbf{V}_t. \quad (3.5.11)$$

Now the expression for ρ_{tt} in the theorem can be obtained by dotting with \mathbf{e}_3 , giving Equation 3.5.2. Isolating γ_{tt} we have

$$\gamma_{tt} = (\Omega_\times^2 + [\Omega_t]_\times)\gamma + \frac{1}{\rho}(2\Omega_\times\Gamma_t^w + \Gamma_{tt}^w + \mathbf{V}_t - 2\rho_t\gamma_t - \rho_{tt}\gamma). \quad (3.5.12)$$

Substituting Equations 3.5.2 and 3.5.3 into the above, we obtain the final expression for γ_{tt} . ■

The Special Case of Fixed Points. The question of how the image of a fixed point moves as the camera moves was studied by Longuet-Higgins and Prazdny [100] and later by Waxman and Ullman [155], giving the velocity γ_t for a fixed point. This calculation also leads to the well-known *epipolar constraint*, the notion of *Essential matrix* [99], and the *continuous epipolar constraint* [104]. Theorem 3.5.1 in the special case of a fixed point gives interesting geometric insight into these classical results. Specifically, setting $\Gamma_t^w = 0$ in first-order computations of Equations 3.5.1 and 3.5.3 results in

$$\begin{cases} \frac{\rho_t}{\rho} = \mathbf{e}_3^\top(\Omega_\times\gamma + \frac{\mathbf{V}}{\rho}) \\ \gamma_t = \Omega_\times\gamma - (\mathbf{e}_3^\top\Omega_\times\gamma)\gamma + \frac{\mathbf{V}}{\rho} - \frac{\mathbf{V}_z}{\rho}\gamma. \end{cases} \quad \text{at } t = 0 \quad (3.5.13)$$

$$\gamma_t = \Omega_\times\gamma - (\mathbf{e}_3^\top\Omega_\times\gamma)\gamma + \frac{\mathbf{V}}{\rho} - \frac{\mathbf{V}_z}{\rho}\gamma. \quad (3.5.14)$$

Essential constraint. To derive the differential epipolar constraint, eliminate ρ from Equation 3.5.14 by first writing out the equation in terms of ξ and η where $\gamma_t = [\xi, \eta, 0]^\top$ and using $\mathbf{e}_3^\top\Omega_\times\gamma = -\Omega_y\xi + \Omega_x\eta$

$$\begin{cases} u - \Omega_y\xi^2 + \Omega_x\xi\eta + \Omega_z\eta - \Omega_y = \frac{1}{\rho}(V_x - \mathbf{V}_z\xi) \\ v + \Omega_x\eta^2 - \Omega_y\xi\eta - \Omega_z\xi + \Omega_x = \frac{1}{\rho}(V_y - \mathbf{V}_z\eta) \end{cases} \quad (3.5.15)$$

and then eliminating ρ , gives

$$(u - \Omega_y\xi^2 + \Omega_x\xi\eta + \Omega_z\eta - \Omega_y)(V_y - \eta\mathbf{V}_z) = (v + \Omega_x\eta^2 - \Omega_y\xi\eta - \Omega_z\xi + \Omega_x)(V_x - \xi\mathbf{V}_z)$$

which is the epipolar constraint for differential motion [104, 161]. A more direct way of deriving the epipolar constraint equation is to eliminate ρ in Equation 3.5.9 with $\Gamma_t^w = 0$ by taking the dot-product with the vector $\mathbf{V} \times \gamma = \mathbf{V}_\times\gamma$, where \mathbf{V}_\times is a skew-symmetric arrangement of \mathbf{V} , and using $\Omega_\times^\top = -\Omega_\times$ gives

$$\rho\gamma_t^\top\mathbf{V}_\times\gamma = -\rho\gamma^\top\Omega_\times\mathbf{V}_\times\gamma$$

resulting in the *differential epipolar constraint*

$$\gamma_t^\top\mathbf{V}_\times\gamma + \gamma^\top\Omega_\times\mathbf{V}_\times\gamma = 0. \quad (3.5.16)$$

In comparison, the widely-known essential constraint for relating two views is given by:

$$\gamma_2^\top \mathcal{T}_\times \mathcal{R} \gamma_1 = 0, \quad (3.5.17)$$

where the matrix $\mathcal{T}_\times \mathcal{R}$, the essential matrix, combines the effects of translation and rotation to relate two points γ_1 and γ_2 . In the differential case the two matrices \mathbf{V}_\times and $\Omega_\times \mathbf{V}_\times$ play a similar role to $\mathcal{T}_\times \mathcal{R}$ in the discrete motion case to relate a point and its velocity.

Remark 3.5.1. Observe from Equation 3.5.14 that γ_t can also be written as the sum of two components, one depending on \mathbf{V} , and the other on Ω , *i.e.*,

$$\gamma_t = \frac{1}{\rho} A(\gamma) \mathbf{V} + B(\gamma) \Omega, \quad \text{where } A(\gamma) = \begin{bmatrix} 1 & 0 & -\xi \\ 0 & 1 & -\eta \\ 0 & 0 & 0 \end{bmatrix} \text{ and } B(\gamma) = \begin{bmatrix} -\xi\eta & 1 + \xi^2 & -\eta \\ -(1 + \eta) & \xi\eta & \xi \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.5.18)$$

That γ_t depends linearly on \mathbf{V} and Ω (since A and B are only dependent on the position γ) is the basis of subspace methods in structure from motion [72], which observe that

$$Q = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ \xi & -\eta & 0 \end{bmatrix}, \quad QAV = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ \xi & -\eta & 0 \end{bmatrix} \mathbf{V} = \mathbf{V} \times \gamma, \quad QB\Omega = \begin{bmatrix} -\xi\eta & 1 + \xi^2 & -\eta \\ -(1 + \eta) & \xi\eta & \xi \\ 0 & 0 & 0 \end{bmatrix} \Omega = \gamma \times (\gamma \times \Omega), \quad (3.5.19)$$

to define transformed velocities q

$$q = Q\gamma_t = \frac{1}{\rho} \mathbf{V} \times \gamma + \gamma \times (\gamma \times \Omega). \quad (3.5.20)$$

This equation provides a linear constraint in \mathbf{V} and Ω , so that observations of image velocities q_1, q_2, \dots, q_N at points $\gamma_1, \gamma_2, \dots, \gamma_N$ provides N constraints of the form

$$q_i = \frac{1}{\rho_i} \mathbf{V} \times \gamma_i + \gamma_i \times (\gamma_i \times \Omega), \quad i = 1, 2, \dots, N. \quad (3.5.21)$$

3.5.2 Differential Relations for a Curve

Theorem 3.5.2. (*Deforming 3D curve*) Consider a deforming 3D curve $\mathbf{\Gamma}(s, t)$ projecting to a family of 2D curves $\gamma(s, t)$ with depth $\rho(s, t)$, arising from camera motion with differential velocities of translation and rotation \mathbf{V} and Ω , respectively, and let \mathbf{V}_t and Ω_t be their respective derivatives with time. Then, the image velocity γ_t is determined from $\left\{ \mathbf{V}, \Omega, \frac{\mathbf{V}}{\rho}, \frac{\mathbf{\Gamma}_t^w}{\rho}, \mathbf{t} \right\}$,

$$\gamma_t = \alpha \mathbf{t} + \beta \mathbf{n}, \quad \text{where } \begin{cases} \alpha = -\Omega \cdot \gamma \times (\gamma \times \mathbf{n}) - \left(\frac{\mathbf{V}}{\rho} - \Omega_\times \frac{\mathcal{T}}{\rho} + \mathcal{R} \frac{\mathbf{\Gamma}_t^w}{\rho} \right) \cdot \gamma \times \mathbf{n}, & (3.5.22) \\ \beta = \Omega \cdot \gamma \times (\gamma \times \mathbf{t}) + \left(\frac{\mathbf{V}}{\rho} - \Omega_\times \frac{\mathcal{T}}{\rho} + \mathcal{R} \frac{\mathbf{\Gamma}_t^w}{\rho} \right) \cdot \gamma \times \mathbf{t}, & (3.5.23) \end{cases}$$

Proof. From Equation 3.2.14 and using $-\mathcal{R}c_t = \mathbf{V} - \Omega_\times \mathcal{T}$ from Equation 3.2.11,

$$\mathbf{\Gamma}_t = \Omega_\times \mathbf{\Gamma} + \mathbf{V} - \Omega_\times \mathcal{T} + \mathcal{R} \mathbf{\Gamma}_t^w. \quad (3.5.24)$$

Using $\mathbf{\Gamma} = \rho\boldsymbol{\gamma}$ and $\mathbf{\Gamma}_t = \rho_t\boldsymbol{\gamma} + \rho\boldsymbol{\gamma}_t$,

$$\rho_t\boldsymbol{\gamma} + \rho\boldsymbol{\gamma}_t = \rho\boldsymbol{\Omega}_\times\boldsymbol{\gamma} + \mathbf{V} - \boldsymbol{\Omega}_\times\mathcal{T} + \mathcal{R}\boldsymbol{\Gamma}_t^w. \quad (3.5.25)$$

Taking the dot product with $\boldsymbol{\gamma} \times \mathbf{n}$ and $\boldsymbol{\gamma} \times \mathbf{t}$

$$\begin{cases} \rho\boldsymbol{\gamma}_t \cdot (\boldsymbol{\gamma} \times \mathbf{n}) = \rho(\boldsymbol{\Omega}_\times\boldsymbol{\gamma}) \cdot (\boldsymbol{\gamma} \times \mathbf{n}) + (\mathbf{V} - \boldsymbol{\Omega}_\times\mathcal{T}) \cdot (\boldsymbol{\gamma} \times \mathbf{n}) + \mathcal{R}\boldsymbol{\Gamma}_t^w \cdot (\boldsymbol{\gamma} \times \mathbf{n}), \\ \rho\boldsymbol{\gamma}_t \cdot (\boldsymbol{\gamma} \times \mathbf{t}) = \rho(\boldsymbol{\Omega}_\times\boldsymbol{\gamma}) \cdot (\boldsymbol{\gamma} \times \mathbf{t}) + (\mathbf{V} - \boldsymbol{\Omega}_\times\mathcal{T}) \cdot (\boldsymbol{\gamma} \times \mathbf{t}) + \mathcal{R}\boldsymbol{\Gamma}_t^w \cdot (\boldsymbol{\gamma} \times \mathbf{t}). \end{cases} \quad (3.5.26)$$

Now,

$$\begin{cases} \boldsymbol{\gamma}_t \cdot (\boldsymbol{\gamma} \times \mathbf{n}) = (\boldsymbol{\alpha}\mathbf{t} + \beta\mathbf{n}) \cdot (\boldsymbol{\gamma} \times \mathbf{n}) = \boldsymbol{\alpha}\mathbf{t} \cdot (\boldsymbol{\gamma} \times \mathbf{n}) = \boldsymbol{\alpha}\mathbf{n} \times \mathbf{t} \cdot \boldsymbol{\gamma} = -\boldsymbol{\alpha}\mathbf{e}_3^\top\boldsymbol{\gamma} = -\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_t \cdot (\boldsymbol{\gamma} \times \mathbf{t}) = (\boldsymbol{\alpha}\mathbf{t} + \beta\mathbf{n}) \cdot (\boldsymbol{\gamma} \times \mathbf{t}) = \beta\mathbf{n} \cdot (\boldsymbol{\gamma} \times \mathbf{t}) = \beta\mathbf{t} \times (\mathbf{n} \cdot \boldsymbol{\gamma}) = \beta\mathbf{e}_3^\top\boldsymbol{\gamma} = \beta. \end{cases} \quad (3.5.27)$$

So that we can write

$$\begin{cases} \boldsymbol{\alpha} = -(\boldsymbol{\Omega}_\times\boldsymbol{\gamma}) \cdot (\boldsymbol{\gamma} \times \mathbf{n}) - \left(\frac{\mathbf{V}}{\rho} - \boldsymbol{\Omega}_\times\frac{\mathcal{T}}{\rho} + \mathcal{R}\frac{\boldsymbol{\Gamma}_t^w}{\rho} \right) \cdot (\boldsymbol{\gamma} \times \mathbf{n}) \\ \boldsymbol{\beta} = (\boldsymbol{\Omega}_\times\boldsymbol{\gamma}) \cdot (\boldsymbol{\gamma} \times \mathbf{t}) + \left(\frac{\mathbf{V}}{\rho} - \boldsymbol{\Omega}_\times\frac{\mathcal{T}}{\rho} + \mathcal{R}\frac{\boldsymbol{\Gamma}_t^w}{\rho} \right) \cdot (\boldsymbol{\gamma} \times \mathbf{t}). \end{cases} \quad (3.5.28)$$

Since we can switch the cross and dot products in a triple scalar product, $\boldsymbol{\Omega}_\times\boldsymbol{\gamma} \cdot (\boldsymbol{\gamma} \times \mathbf{n}) = \boldsymbol{\Omega} \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{n})$ and $\boldsymbol{\Omega}_\times\boldsymbol{\gamma} \cdot (\boldsymbol{\gamma} \times \mathbf{t}) = \boldsymbol{\Omega} \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})$, giving the final result. ■

Corollary 3.5.3. *The spatial variation of the velocity vector field $\boldsymbol{\gamma}_t$ along the curve and in time can be written as*

$$\begin{aligned} \boldsymbol{\gamma}_{st} &= (-\mathbf{V} + V_z\boldsymbol{\gamma}) \frac{\rho_s}{\rho^2} - \frac{V_z}{\rho}\boldsymbol{\gamma}_s + \boldsymbol{\Omega}_\times\boldsymbol{\gamma}_s - (\mathbf{e}_3^\top\boldsymbol{\Omega}_\times\boldsymbol{\gamma}_s)\boldsymbol{\gamma} - (\mathbf{e}_3^\top\boldsymbol{\Omega}_\times\boldsymbol{\gamma})\boldsymbol{\gamma}_s \\ &\quad - \frac{1}{\rho^2}(\boldsymbol{\Gamma}_{st}^w - \mathbf{e}_3^\top\boldsymbol{\Gamma}_{st}^w\boldsymbol{\gamma} - \mathbf{e}_3^\top\boldsymbol{\Gamma}_t^w\boldsymbol{\gamma}_s) - \frac{1}{\rho^2}(\boldsymbol{\Gamma}_t^w - \mathbf{e}_3^\top\boldsymbol{\Gamma}_t^w\boldsymbol{\gamma})\rho_s \end{aligned} \quad (3.5.29)$$

and the time acceleration $\boldsymbol{\gamma}_{tt}$ is defined by

$$\begin{cases} \mathbf{t}^\top\boldsymbol{\gamma}_{tt} = \mathbf{t}^\top(\boldsymbol{\Omega}_\times^2 + [\boldsymbol{\Omega}_t]_\times)\boldsymbol{\gamma} + \frac{2}{\rho}\mathbf{t}^\top\boldsymbol{\Omega}_\times\boldsymbol{\Gamma}_t^w + \frac{1}{\rho}\mathbf{t}^\top\boldsymbol{\Gamma}_{tt}^w + \frac{1}{\rho}\mathbf{t}^\top\mathbf{V}_t \\ \quad - 2\mathbf{e}_3^\top\left(\boldsymbol{\Omega}_\times\boldsymbol{\gamma} + \frac{\mathbf{V}}{\rho} + \frac{\boldsymbol{\Gamma}_t^w}{\rho}\right)\boldsymbol{\alpha} - \mathbf{e}_3^\top\left((\boldsymbol{\Omega}_\times^2 + [\boldsymbol{\Omega}_t]_\times)\boldsymbol{\gamma} + \frac{\mathbf{V}_t}{\rho} + 2\boldsymbol{\Omega}_\times\frac{\boldsymbol{\Gamma}_t^w}{\rho} + \frac{\boldsymbol{\Gamma}_{tt}^w}{\rho}\right)\mathbf{t}^\top\boldsymbol{\gamma}, \\ \mathbf{n}^\top\boldsymbol{\gamma}_{tt} = \mathbf{n}^\top(\boldsymbol{\Omega}_\times^2 + [\boldsymbol{\Omega}_t]_\times)\boldsymbol{\gamma} + \frac{2}{\rho}\mathbf{n}^\top\boldsymbol{\Omega}_\times\boldsymbol{\Gamma}_t^w + \frac{1}{\rho}\mathbf{n}^\top\boldsymbol{\Gamma}_{tt}^w + \frac{1}{\rho}\mathbf{n}^\top\mathbf{V}_t \\ \quad - 2\mathbf{e}_3^\top\left(\boldsymbol{\Omega}_\times\boldsymbol{\gamma} + \frac{\mathbf{V}}{\rho} + \frac{\boldsymbol{\Gamma}_t^w}{\rho}\right)\boldsymbol{\beta} - \mathbf{e}_3^\top\left((\boldsymbol{\Omega}_\times^2 + [\boldsymbol{\Omega}_t]_\times)\boldsymbol{\gamma} + \frac{\mathbf{V}_t}{\rho} + 2\boldsymbol{\Omega}_\times\frac{\boldsymbol{\Gamma}_t^w}{\rho} + \frac{\boldsymbol{\Gamma}_{tt}^w}{\rho}\right)\mathbf{n}^\top\boldsymbol{\gamma}, \end{cases} \quad (3.5.30)$$

Proof. The $\boldsymbol{\gamma}_{st}$ expression in (3.5.29) is derived by differentiating $\boldsymbol{\gamma}_t$ with respect to s in Equation 3.5.3, but notice that $\boldsymbol{\gamma}_t$ in the moving case decomposes into the same terms as for the fixed case, Equation 3.5.6, plus terms dependent on $\boldsymbol{\Gamma}_t^w$ given by $\frac{1}{\rho}(\boldsymbol{\Gamma}_t^w - \mathbf{e}_3^\top\boldsymbol{\Gamma}_t^w\boldsymbol{\gamma})$. Differentiating with respect to s then gives a term equal to $\boldsymbol{\gamma}_{st}$ for the fixed case plus terms dependent on $\boldsymbol{\Gamma}_t^w$ and its

spatial derivative, the latter being obtained by differentiating the above expression with respect to s .

The expressions of γ_{tt} in the Frenet frame were obtained by taking the dot product of (3.5.4) with \mathbf{t} and \mathbf{n} , noting that $\gamma_t \cdot \mathbf{t} = \alpha$ and $\gamma_t \cdot \mathbf{n} = \beta$. We then substitute the expressions for ρ_t and ρ_{tt} in. ■

Special Case: Rigid Stationary Curve.

Corollary 3.5.4. (*Rigid stationary 3D curve*) Let $\Gamma(\tilde{s})$ be a 3D curve projecting to a family of 2D curves $\gamma(\tilde{s}, t)$ with depth $\rho(\tilde{s}, t)$, arising from camera motion with differential velocity of translation and rotation \mathbf{V} and Ω , respectively. Let \mathbf{t} denote the unit tangent to the image curve. Then

$$\gamma_{\tilde{s}t} = \frac{-\rho_{\tilde{s}}}{\rho} \left(\frac{\mathbf{V}}{\rho} - \frac{V_z}{\rho} \gamma \right) - \frac{V_z}{\rho} \mathbf{t} + \Omega \times \mathbf{t} - (\mathbf{e}_3^\top \Omega \times \mathbf{t}) \gamma - (\mathbf{e}_3^\top \Omega \times \gamma) \mathbf{t}. \quad (3.5.31)$$

Proof. Follows by setting $\Gamma_t^w = 0$ in Equation 3.5.29 and using the spatial parameter as the arclength of the image curve. ■

Corollary 3.5.5. *The tangential and normal velocities of a rigid curve induced by a moving camera are derived from $\{\gamma, \mathbf{t}, \mathbf{n}, \frac{\mathcal{T}}{\rho}, \Omega, \frac{\mathbf{V}}{\rho}\}$ for any t as*

$$\begin{cases} \alpha = -\Omega \cdot \gamma \times (\gamma \times \mathbf{n}) - \left(\frac{\mathbf{V}}{\rho} - \Omega \times \frac{\mathcal{T}}{\rho} \right) \cdot (\gamma \times \mathbf{n}) \\ \beta = \Omega \cdot \gamma \times (\gamma \times \mathbf{t}) + \left(\frac{\mathbf{V}}{\rho} - \Omega \times \frac{\mathcal{T}}{\rho} \right) \cdot (\gamma \times \mathbf{t}) \end{cases} \quad \text{for any } t, \quad (3.5.32)$$

$$\quad (3.5.33)$$

or

$$\begin{cases} \alpha = -\Omega \cdot \gamma \times (\gamma \times \mathbf{n}) - \gamma \times \mathbf{n} \cdot \frac{\mathbf{V}}{\rho} \\ \beta = \Omega \cdot \gamma \times (\gamma \times \mathbf{t}) + \gamma \times \mathbf{t} \cdot \frac{\mathbf{V}}{\rho} \end{cases} \quad \text{for } t = 0. \quad (3.5.34)$$

$$\quad (3.5.35)$$

Corollary 3.5.6. *The infinitesimal Essential constraint in the Frenet frame of the image of a rigid curve is given by:*

$$(\gamma \times \mathbf{t}) \cdot \mathbf{V} [\alpha + \Omega \cdot \gamma \times (\gamma \times \mathbf{n})] + (\gamma \times \mathbf{n}) \cdot \mathbf{V} [\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})] = 0 \quad (3.5.36)$$

Proof. Eliminate ρ from (3.5.35) and (3.5.34). ■

Corollary 3.5.7. (*From [120, 122]*) *The tangential velocity α can be fully determined from the normal velocity β and $\gamma, \mathbf{t}, \mathbf{n}, \Omega$, and $\frac{\mathbf{V}}{\rho}$ without the explicit knowledge of ρ , as*

$$\alpha = -[\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})] \frac{\mathbf{V} \cdot (\gamma \times \mathbf{n})}{\mathbf{V} \cdot (\gamma \times \mathbf{t})} - \Omega \cdot \gamma \times (\gamma \times \mathbf{n}) \quad (3.5.37)$$

Proof. Follows by eliminating ρ from Equations 3.5.34 and 3.5.35. ■

Special Case: Occluding Contours. A remarkable observation is derived below that the first-order deformation of an apparent contour under epipolar parametrization does not depend on the 3D surface geometry, since the curvature-dependent terms cancel out for an occluding contour.

Theorem 3.5.8. (*Occluding contours*) Let $\Gamma(s, t)$ be the contour generator for apparent contours $\gamma(s, t)$. Then the image velocity γ_t at $t = 0$ can be determined from γ by ρ and the infinitesimal motion parameters using Equation 3.5.14, i.e., the same one used for a stationary contour.

Proof. Recall from Equation 3.2.22 that the velocity of an occluding contour under epipolar parametrization satisfies $\Gamma_t^w = \lambda(\Gamma^w - \mathbf{c})$ for some λ , so that at $t = 0$

$$\Gamma_t^w = \lambda\rho\gamma \implies \mathbf{e}_3^\top \Gamma_t^w = \lambda\rho, \quad (3.5.38)$$

so that $\Gamma_t^w = (\mathbf{e}_3^\top \Gamma_t^w)\gamma$ and the terms $\Gamma_t^w - (\mathbf{e}_3^\top \Gamma_t^w)\gamma = 0$ so that all appearances of Γ_t^w cancel-out altogether in Equation 3.5.3, giving exactly the same formula as for fixed contours, Equation 3.5.14, when $\Gamma_t^w = 0$. ■

We now show exactly how the velocity of the 3D occluding contour, Γ_t^w , depends on the curvature of the occluding surface.

Theorem 3.5.9. *The velocity of a 3D occluding contour under epipolar parametrization and relative to a fixed world coordinate system (camera at $t = 0$) is given by:*

$$\begin{cases} \Gamma_t^w = -\frac{\mathbf{c}_t^\top \mathbf{N}^w}{K^t} \cdot \frac{\Gamma^w - \mathbf{c}}{\|\Gamma^w - \mathbf{c}\|^2}, & \text{for arbitrary } t. \\ \Gamma_t^w = -\frac{\mathbf{c}_t^\top \mathbf{N}}{K^t} \cdot \frac{\gamma}{\rho\|\gamma\|^2}, & \text{for } t = 0, \end{cases} \quad (3.5.39)$$

$$\Gamma_t^w = -\frac{\mathbf{c}_t^\top \mathbf{N}}{K^t} \cdot \frac{\gamma}{\rho\|\gamma\|^2}, \quad \text{for } t = 0, \quad (3.5.40)$$

or, in terms of \mathcal{T} and R , and image measurements:

$$\Gamma_t^w = \frac{1}{K^t} \left(\frac{\mathbf{V}^\top}{\rho} \frac{\gamma \times \mathbf{t}}{\|\gamma \times \mathbf{t}\|} \right) \frac{\gamma}{\|\gamma\|^2}, \quad \text{for } t = 0, \quad (3.5.41)$$

where K^t is the normal curvature of the occluding surface along the visual direction.

Proof. Differentiating the occluding contour condition in the second form of Equation 3.2.21 gives

$$(\Gamma_t^w - \mathbf{c}_t)^\top \mathbf{N}^w + (\Gamma^w - \mathbf{c})^\top \mathbf{N}_t^w = 0. \quad (3.5.42)$$

The Weingarten equations of differential geometry relate sectional curvature in a direction to the derivative of the normal in that direction, $\mathbf{N}_t^w = -K^t \Gamma_t^w$, so that using $\Gamma_t^{w\top} \mathbf{N}^w = 0$

$$(\Gamma_t^w - \mathbf{c}_t)^\top \mathbf{N}^w - K^t (\Gamma^w - \mathbf{c})^\top \Gamma_t^w = 0. \quad (3.5.43)$$

$$-\mathbf{c}_t^\top \mathbf{N}^w - K^t (\Gamma^w - \mathbf{c})^\top \Gamma_t^w = 0 \quad (3.5.44)$$

$$(\Gamma^w - \mathbf{c})^\top \Gamma_t^w = -\frac{\mathbf{c}_t^\top \mathbf{N}^w}{K^t}. \quad (3.5.45)$$

Now, for the *epipolar parametrization*, $\Gamma_t^w = \lambda(\Gamma^w - \mathbf{c})$, so that λ can be obtained leading to

$$\Gamma_t^w = -\frac{\mathbf{c}_t^\top \mathbf{N}^w}{K^t} \frac{\Gamma^w - \mathbf{c}}{\|\Gamma^w - \mathbf{c}\|^2}, \quad \text{for arbitrary } t. \quad (3.5.46)$$

At $t = 0$, we have $\mathbf{N}^w = \mathbf{N}$ and $\mathbf{\Gamma}^w - \mathbf{c} = \mathbf{\Gamma} = \rho\boldsymbol{\gamma}$ (but note that $\mathbf{\Gamma}_t(0) \neq \mathbf{\Gamma}_t^w(0)$), hence:

$$\mathbf{\Gamma}_t^w = -\frac{\mathbf{c}_t^\top \mathbf{N}}{K^t} \frac{\boldsymbol{\gamma}}{\rho \|\boldsymbol{\gamma}\|^2}, \quad \text{for } t = 0. \quad (3.5.47)$$

Using $\mathbf{V} = -\mathbf{c}_t$ from Equation 3.2.11, and $\mathbf{N} = \frac{\boldsymbol{\gamma} \times \mathbf{t}}{\|\boldsymbol{\gamma} \times \mathbf{t}\|}$ from Section 3.2.5 instead of \mathbf{N} gives the alternative form of this equation. ■

We now present a theorem relating observed quantities to camera motion. A form of this theorem appears in [120, 122], Equation L1, but this is limited to rigid motion. Furthermore, these results are incorrect in that they missed a term corresponding to the last term in Equation 3.5.48,

$$(\Omega_\times \mathbf{V})(\boldsymbol{\gamma} \times \mathbf{t})[\beta - \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})],$$

perhaps because they differentiated the normal flow equation for $t = 0$. The theorem below corrects that and also generalizes the results to include occluding contours. The fact that Equation 3.5.48 in the theorem is also valid for occluding contours is a new result, to the best of our knowledge. The term $\mathbf{\Gamma}_t^w$ is zero for fixed contours, and is dependent on surface curvature in the case of occluding contours. The equation is not valid for arbitrary non-rigid contours because, in order to derive the normal flow equation, we used $\mathbf{\Gamma}_t^w \cdot (\boldsymbol{\gamma} \times \mathbf{t}) = 0$, which is only true for occluding and fixed contours.

Theorem 3.5.10. *(A corrected and generalized form of the L1 equation of [120, 122]) Given a 3D occluding contour or fixed curve, and the family of projected curves $\boldsymbol{\gamma}(t)$ observed in a monocular sequence of images from a moving camera, and given $\mathbf{t}, \kappa, \mathbf{n}, \beta, \beta_t$ measurements at one point, then the first and second order camera motion, $\Omega, \mathbf{V}, \Omega_t, \mathbf{V}_t$ satisfy a polynomial equation:*

$$\begin{aligned} & V_z [\beta - \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})]^2 + \mathbf{V} \cdot \boldsymbol{\gamma} \times \mathbf{t} (\beta_t - \Omega_t \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t}) - \Omega \cdot [\boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})]_t) \\ & - [\mathbf{V}_t \cdot \boldsymbol{\gamma} \times \mathbf{t} - \mathbf{V} \cdot (\boldsymbol{\gamma} \times \mathbf{t})_t] [\beta - \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})] + \mathbf{V} \cdot \boldsymbol{\gamma} \times \mathbf{t} (\mathbf{e}_3 \cdot \Omega_\times \boldsymbol{\gamma}) [\beta - \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})] \\ & + \mathbf{e}_3 \cdot \mathbf{\Gamma}_t^w [\beta - \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})]^2 + (\Omega_\times \mathbf{V})(\boldsymbol{\gamma} \times \mathbf{t}) [\beta - \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})] = 0. \end{aligned} \quad (3.5.48)$$

Proof. The normal velocity β of an image contour follows Equation (3.5.33), which holds for both stationary curves, Corollary 3.5.5, and for occluding contours, Theorem 3.5.8. Differentiating it with respect to time,

$$\begin{aligned} \rho_t \beta + \beta_t \rho = & \rho_t \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t}) + \rho \Omega_t \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t}) + \rho \Omega [\boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})]_t \\ & + (\boldsymbol{\gamma} \times \mathbf{t})_t (\mathbf{V} - \Omega_\times \mathcal{T}) + (\boldsymbol{\gamma} \times \mathbf{t}) (\mathbf{V}_t - \Omega_t \times \mathcal{T} - \Omega_\times \mathbf{V}) \end{aligned} \quad (3.5.49)$$

Rearranging the terms,

$$\begin{aligned} & \rho_t [\beta - \Omega \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})] + \rho [\beta_t - \Omega_t \cdot \boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t}) - \Omega \cdot [\boldsymbol{\gamma} \times (\boldsymbol{\gamma} \times \mathbf{t})]_t] \\ & = (\boldsymbol{\gamma} \times \mathbf{t})_t (\mathbf{V} - \Omega_\times \mathcal{T}) + (\boldsymbol{\gamma} \times \mathbf{t}) (\mathbf{V}_t - \Omega_t \times \mathcal{T} - \Omega_\times \mathbf{V}). \end{aligned} \quad (3.5.50)$$

Setting $t = 0$,

$$\begin{aligned} & \rho_t[\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})] + \rho[\beta_t + -\Omega_t \cdot \gamma \times (\gamma \times \mathbf{t}) - \Omega \cdot [\gamma \times (\gamma \times \mathbf{t})]_t] \\ & = (\gamma \times \mathbf{t})_t \mathbf{V} + (\gamma \times \mathbf{t})(\mathbf{V}_t - \Omega_\times \mathbf{V}) \end{aligned} \quad (3.5.51)$$

Now, from Equation 3.5.1, we can plug-in an expression for ρ_t at $t = 0$:

$$\begin{aligned} & (\rho \mathbf{e}_3^\top \Omega_\times \gamma + V_z + \mathbf{e}_3^\top \Gamma_t^w)[\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})] + \rho(\beta_t - \Omega_t \cdot \gamma \times (\gamma \times \mathbf{t}) - \Omega \cdot [\gamma \times (\gamma \times \mathbf{t})]_t) \\ & = (\gamma \times \mathbf{t})_t \mathbf{V} + (\gamma \times \mathbf{t})(\mathbf{V}_t - \Omega_\times \mathbf{V}), \end{aligned} \quad (3.5.52)$$

which is analogous to Equation 7.28 of [120, p.167], but this time with corrected temporal derivatives and with occluding contours also being included. Now, eliminating depth ρ using Equation 3.5.35, *e.g.*, by multiplying the above by $[\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})]$, we obtain:

$$\begin{aligned} & [V_z(\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})) + (\mathbf{V} \cdot \gamma \times \mathbf{t}) \mathbf{e}_3 \cdot \Omega_\times \gamma + \\ & \mathbf{e}_3 \cdot \Gamma_t^w(\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t}))][\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})] \\ & + \mathbf{V} \cdot \gamma \times \mathbf{t} [\beta_t - \Omega_t \cdot \gamma \times (\gamma \times \mathbf{t}) - \Omega \cdot [\gamma \times (\gamma \times \mathbf{t})]_t] = \\ & [\mathbf{V}_t \cdot \gamma \times \mathbf{t} + \mathbf{V}(\gamma \times \mathbf{t})_t - (\Omega_\times \mathbf{V})(\gamma \times \mathbf{t})][\beta - \Omega \cdot \gamma \times (\gamma \times \mathbf{t})]. \end{aligned} \quad (3.5.53)$$

Rearranging the terms, we obtain the desired equation. ■

Theorem 3.5.11. *The first spatial derivative of image apparent motion of both a fixed curve and an occluding contour under epipolar correspondence is given by:*

$$\gamma_{st} = \left(-\frac{\mathbf{V}}{\rho} + \mathbf{e}_3^\top \frac{\mathbf{V}}{\rho} \gamma \right) \frac{\rho_s}{\rho} - \mathbf{e}_3^\top \frac{\mathbf{V}}{\rho} \gamma_s + \Omega_\times \gamma_s - (\mathbf{e}_3^\top \Omega_\times \gamma_s) \gamma - (\mathbf{e}_3^\top \Omega_\times \gamma) \gamma_s, \quad (3.5.54)$$

which is parametrized by rotational and depth-normalized translational velocities (5 parameters), depth-normalized spatial derivative of depth (1 parameter), a total of 7 parameters. Note that the derivative of depth ρ_s can be expressed in terms of 3D curve geometry as $\rho_s = \mathbf{e}_3^\top \Gamma_s$

Proof. Equation 3.5.54 follows by differentiating the fixed flow (3.5.14) with respect to s , observing that only ρ and γ depend on s . The formula for ρ_s is obtained from the observation that the dot product of $\Gamma = \rho \gamma$ with \mathbf{e}_3 gives $\mathbf{e}_3^\top \Gamma = \rho$. Differentiating this with respect to s gives $\rho_s = \mathbf{e}_3^\top \Gamma_s$. ■

Theorem 3.5.1 gives an expression for the image acceleration of a moving 3D point, which includes points lying on any type of contour (even non-rigid), in terms of the evolution of the 3D curve. Since the latter is expressed in terms of a fixed world coordinate system, the motion of the object and the motion of the cameras are written down separately, even though they exert joint effects on image velocity.

Theorem 3.5.12. *The image acceleration of an occluding contour under epipolar parametrization is given by:*

$$\begin{aligned} \gamma_{tt} & = (\Omega_\times^2 + [\Omega_t]_\times) \gamma - \mathbf{e}_3^\top (\Omega_\times^2 + [\Omega_t]_\times) \gamma + 2\Omega_\times \frac{\Gamma_t^w}{\rho} + \frac{\mathbf{V}_t}{\rho} - \frac{2\rho_t}{\rho} \gamma_t + \frac{\mathbf{e}_3^\top \Gamma_t^w}{\rho} \gamma_t \\ & - \frac{\mathbf{e}_3^\top \Gamma_t^w}{\rho} \Omega_\times \gamma - \frac{\mathbf{e}_3^\top \mathbf{V}_t}{\rho} \gamma - \frac{2\mathbf{e}_3^\top \Omega_\times \Gamma_t^w}{\rho} \gamma, \quad \text{at } t = 0, \end{aligned} \quad (3.5.55)$$

where γ_t and ρ_t are given by Equations 3.5.14 and 3.5.13, and $\mathbf{\Gamma}_t^w$ is dependent on curvature, Equation 3.5.41.

Proof. The first part of the proof is the same as the proof of Theorem 3.5.1. Substituting Equation 3.5.2 into Equation 3.5.12 we get:

$$\gamma_{tt} = (\Omega_{\times}^2 + [\Omega_t]_{\times})\gamma + \frac{2\Omega_{\times}\mathbf{\Gamma}_t^w}{\rho} + \frac{\mathbf{\Gamma}_{tt}^w}{\rho} + \frac{\mathbf{V}_t}{\rho} - \frac{2\rho_t\gamma_t}{\rho} - [e_3^{\top}(\Omega_{\times}^2 + [\Omega_t]_{\times})\gamma]\gamma - \frac{e_3^{\top}\mathbf{V}_t}{\rho}\gamma - \frac{2e_3^{\top}\Omega_{\times}\mathbf{\Gamma}_t^w}{\rho}\gamma - \frac{e_3^{\top}\mathbf{\Gamma}_{tt}^w}{\rho}\gamma. \quad (3.5.56)$$

Now, let \mathbf{v} be the viewing direction in world coordinates, so that:

$$\gamma = R\mathbf{v}, \quad (3.5.57)$$

and let \mathbf{f} be the normal to the image plane in world coordinates, so that

$$e_3 = R\mathbf{f}. \quad (3.5.58)$$

Thus, we have that

$$e_3^{\top}\gamma = \mathbf{f}^{\top}R^{\top}R\mathbf{v} = \mathbf{f}^{\top}\mathbf{v} = 1. \quad (3.5.59)$$

Note also that at $t = 0$ we have $\mathbf{f} = e_3$ and $\gamma = \mathbf{v}$. Now, the condition for epipolar parametrization of an occluding contour, Equation 3.2.22, can be expressed as:

$$\mathbf{\Gamma}_t^w = \lambda\mathbf{v}, \quad (3.5.60)$$

for some constant λ . Taking the dot product with \mathbf{f} we have:

$$\lambda = \mathbf{f}^{\top}\mathbf{\Gamma}_t^w, \quad (3.5.61)$$

and, substituting back we have

$$\mathbf{\Gamma}_t^w = \mathbf{f}^{\top}\mathbf{\Gamma}_t^w\mathbf{v}. \quad (3.5.62)$$

Differentiating with respect to time gives:

$$\mathbf{\Gamma}_{tt}^w = \lambda_t\mathbf{v} + \lambda\mathbf{v}_t = \lambda_t\mathbf{v} + \mathbf{f}^{\top}\mathbf{\Gamma}_t^w\mathbf{v}_t. \quad (3.5.63)$$

Taking the dot product with \mathbf{f} :

$$\mathbf{f}^{\top}\lambda_t\mathbf{v} = \mathbf{f}^{\top}\mathbf{\Gamma}_{tt}^w\lambda_t = \mathbf{f}^{\top}\mathbf{\Gamma}_{tt}^w. \quad (3.5.64)$$

Thus

$$\mathbf{\Gamma}_{tt}^w = \mathbf{f}^{\top}\mathbf{\Gamma}_{tt}^w\mathbf{v} + \mathbf{f}^{\top}\mathbf{\Gamma}_t^w\mathbf{v}_t, \quad (3.5.65)$$

and at $t = 0$ we have

$$\mathbf{\Gamma}_{tt}^w(0) = e_3\mathbf{\Gamma}_{tt}^w\gamma + e_3^{\top}\mathbf{\Gamma}_t^w\mathbf{v}_t(0). \quad (3.5.66)$$

In order to get $\mathbf{v}_t(0)$ in terms of γ we write

$$\gamma_t = R_t\mathbf{v} + R\mathbf{v}_t, \quad (3.5.67)$$

thus

$$\mathbf{v}_t = \gamma_t - \Omega \times \gamma. \quad (3.5.68)$$

Substituting back into (3.5.66), we have

$$\Gamma_{tt}^w = \mathbf{e}_3^\top \Gamma_{tt}^w \gamma + \mathbf{e}_3^\top \Gamma_t^w - \mathbf{e}_3^\top \Gamma_t^w \Omega \times \gamma. \quad (3.5.69)$$

Plugging this equation onto (3.5.56), the $\mathbf{e}_3 \gamma_{tt}^w / \rho$ terms cancel out, giving the final equation. ■

3.6 Projecting Differential Geometry of Surfaces

Previous sections related the differential geometry of moving points and deforming curves with differential properties of their projected image onto a moving camera. This section addresses the relationship between the differential geometry of a surface and the differential geometry of its projection, namely image velocities optical flow and accelerations.

Theorem 3.6.1. (Spatial variation of the image velocity field in terms of surface). *Consider an image point γ arising from Γ lying on a twice-differentiable surface with normal \mathbf{N} and having Weingarten map \mathbf{K}_c in camera coordinates, under differential camera motion Ω, \mathbf{V} . Then the image velocity $\gamma_t = (u, v, 1)^\top$ has first-order and second-order spatial variation*

$$\begin{cases} \gamma_{\xi t} = \left(-\frac{\mathbf{V}}{\rho} + \frac{V_z}{\rho} \gamma\right) \frac{\rho_\xi}{\rho} - \frac{V_z}{\rho} \mathbf{e}_1 + \Omega \times \mathbf{e}_1 + \Omega_y \gamma - (\mathbf{e}_3^\top \Omega \times \gamma) \mathbf{e}_1 \\ \gamma_{\eta t} = \left(-\frac{\mathbf{V}}{\rho} + \frac{V_z}{\rho} \gamma\right) \frac{\rho_\eta}{\rho} - \frac{V_z}{\rho} \mathbf{e}_2 + \Omega \times \mathbf{e}_2 - \Omega_x \gamma - (\mathbf{e}_3^\top \Omega \times \gamma) \mathbf{e}_2. \end{cases} \quad (3.6.1)$$

$$\begin{cases} u_{\xi\xi} = (-V_x + V_z \xi) \frac{\rho_{\xi\xi}}{\rho^2} - 2(-V_x + V_z \xi) \frac{\rho_\xi^2}{\rho^3} + V_z \frac{\rho_\xi}{\rho^2} + 2\Omega_y \\ u_{\xi\eta} = (-V_x + V_z \xi) \frac{\rho_{\xi\eta}}{\rho^2} - 2(-V_x + V_z \xi) \frac{\rho_\xi \rho_\eta}{\rho^3} + V_z \frac{\rho_\eta}{\rho^2} + \Omega_x \\ u_{\eta\eta} = (-V_x + V_z \xi) \frac{\rho_{\eta\eta}}{\rho^2} - 2(-V_x + V_z \xi) \frac{\rho_\eta^2}{\rho^3} \\ v_{\xi\xi} = (-V_y + V_z \eta) \frac{\rho_{\xi\xi}}{\rho^2} - 2(-V_y + V_z \eta) \frac{\rho_\xi^2}{\rho^3} \\ v_{\xi\eta} = (-V_y + V_z \eta) \frac{\rho_{\xi\eta}}{\rho^2} - 2(-V_y + V_z \eta) \frac{\rho_\xi \rho_\eta}{\rho^3} + V_z \frac{\rho_\xi}{\rho^2} + \Omega_y \\ v_{\eta\eta} = (-V_y + V_z \eta) \frac{\rho_{\eta\eta}}{\rho^2} - 2(-V_y + V_z \eta) \frac{\rho_\eta^2}{\rho^3} + V_z \frac{\rho_\eta}{\rho^2} + 2\Omega_x \end{cases} \quad (3.6.2)$$

Proof. For the first-order image motion field, use the general Equation 3.5.54 for an image curve given in Theorem 3.5.11, where here the image curve is taken to be along one of the coordinate axes, so that the parameter s can be taken as ξ or η . The formula admits the specific form given in this theorem by observing that $\gamma_\xi = \mathbf{e}_1$ and $\gamma_\eta = \mathbf{e}_2$. ■

Theorem 3.6.2. (relating depth derivatives to differential geometry of the surface) *The first order relation states that the normalized depth gradient is related to the surface normal by*

$$\frac{\nabla \rho}{\rho} = -\frac{(\mathbf{N}_x, \mathbf{N}_y)}{\mathbf{N} \cdot \boldsymbol{\gamma}}, \quad \mathbf{N} = \frac{\frac{\rho_\xi}{\rho} \mathbf{e}_1 + \frac{\rho_\eta}{\rho} \mathbf{e}_2 - \left(\frac{\rho_\xi}{\rho} \xi + \frac{\rho_\eta}{\rho} \eta + 1\right) \mathbf{e}_3}{\sqrt{\left(\frac{\rho_\xi}{\rho}\right)^2 + \left(\frac{\rho_\eta}{\rho}\right)^2 + \left(\frac{\rho_\xi}{\rho} \xi + \frac{\rho_\eta}{\rho} \eta + 1\right)^2}}. \quad (3.6.3)$$

when $\mathbf{N} \cdot \boldsymbol{\gamma} \neq 0$, i.e., when the point does not lie on an occluding contour. For an occluding contour point the depth gradient blows up.

Proof. The surface can be described by $\mathbf{\Gamma}(\xi, \eta) = \rho \boldsymbol{\gamma} = [\rho \xi, \rho \eta, \rho]^\top$. Differentiating $\mathbf{\Gamma} = \rho \boldsymbol{\gamma}$ we have

$$\begin{cases} \mathbf{\Gamma}_\xi = \rho_\xi \boldsymbol{\gamma} + \rho \boldsymbol{\gamma}_\xi = \rho_\xi \boldsymbol{\gamma} + \rho \mathbf{e}_1 \\ \mathbf{\Gamma}_\eta = \rho_\eta \boldsymbol{\gamma} + \rho \boldsymbol{\gamma}_\eta = \rho_\eta \boldsymbol{\gamma} + \rho \mathbf{e}_2. \end{cases}$$

The vectors $\mathbf{\Gamma}_\xi$ and $\mathbf{\Gamma}_\eta$ span the tangent plane to the surface. We can isolate ρ_ξ and ρ_η by taking the scalar product with surface normal \mathbf{N}

$$\begin{cases} 0 = \rho_\xi \mathbf{N} \cdot \boldsymbol{\gamma} + \rho \mathbf{e}_1 \cdot \mathbf{N} \\ 0 = \rho_\eta \mathbf{N} \cdot \boldsymbol{\gamma} + \rho \mathbf{e}_2 \cdot \mathbf{N}, \end{cases} \quad (3.6.4)$$

which gives

$$\begin{cases} \rho_\xi = -\frac{\rho}{\mathbf{N} \cdot \boldsymbol{\gamma}} \mathbf{N}_x \\ \rho_\eta = -\frac{\rho}{\mathbf{N} \cdot \boldsymbol{\gamma}} \mathbf{N}_y, \end{cases} \quad (3.6.5)$$

deriving the expression for $\frac{\nabla \rho}{\rho}$. Observe that $\nabla \rho = (\rho_\xi, \rho_\eta)$ is invariant to the direction of \mathbf{N} . The surface normal \mathbf{N} can be obtained from $\mathbf{N} = -\frac{\mathbf{\Gamma}_\xi \times \mathbf{\Gamma}_\eta}{\|\mathbf{\Gamma}_\xi \times \mathbf{\Gamma}_\eta\|}$. First,

$$\begin{aligned} \mathbf{\Gamma}_\xi \times \mathbf{\Gamma}_\eta &= (\rho_\xi \boldsymbol{\gamma} + \rho \mathbf{e}_1) \times (\rho_\eta \boldsymbol{\gamma} + \rho \mathbf{e}_2) \\ &= \rho \rho_\xi \boldsymbol{\gamma} \times \mathbf{e}_2 + \rho \rho_\eta \mathbf{e}_1 \times \boldsymbol{\gamma} + \rho^2 \mathbf{e}_3 \\ &= \rho \rho_\xi (\xi \mathbf{e}_3 - \mathbf{e}_1) + \rho \rho_\eta (\eta \mathbf{e}_3 - \mathbf{e}_2) + \rho^2 \mathbf{e}_3 \\ &= \rho^2 \left[-\frac{\rho_\xi}{\rho} \mathbf{e}_1 - \frac{\rho_\eta}{\rho} \mathbf{e}_2 + \left(\frac{\rho_\xi}{\rho} \xi + \frac{\rho_\eta}{\rho} \eta + 1\right) \mathbf{e}_3 \right], \end{aligned} \quad (3.6.6)$$

which after normalizing gives the result. ■

Theorem 3.6.3. *The Hessian of depth is given by*

$$\begin{cases} \rho_{\xi\xi} = \frac{2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho^2 \mathbf{N}_x^2}{(\mathbf{N} \cdot \boldsymbol{\gamma})^3} + \frac{[4\rho \boldsymbol{\gamma}^\top \mathbf{K}_c \mathbf{e}_1 + 2\mathbf{N}_x] \rho \mathbf{N}_x}{(\mathbf{N} \cdot \boldsymbol{\gamma})^2} + \frac{2\mathbf{e}_1^\top \mathbf{K}_c \mathbf{e}_1 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \\ \rho_{\eta\eta} = \frac{2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho^2 \mathbf{N}_y^2}{(\mathbf{N} \cdot \boldsymbol{\gamma})^3} + \frac{[4\rho \boldsymbol{\gamma}^\top \mathbf{K}_c \mathbf{e}_2 + 2\mathbf{N}_y] \rho \mathbf{N}_y}{(\mathbf{N} \cdot \boldsymbol{\gamma})^2} + \frac{2\mathbf{e}_2^\top \mathbf{K}_c \mathbf{e}_2 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \\ \rho_{\xi\eta} = \frac{2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho^2 \mathbf{N}_x \mathbf{N}_y}{(\mathbf{N} \cdot \boldsymbol{\gamma})^3} + \frac{[2\rho \boldsymbol{\gamma}^\top \mathbf{K}_c \mathbf{e}_2 + \mathbf{N}_y] \rho \mathbf{N}_x + [2\rho \boldsymbol{\gamma}^\top \mathbf{K}_c \mathbf{e}_1 + \mathbf{N}_x] \rho \mathbf{N}_y}{(\mathbf{N} \cdot \boldsymbol{\gamma})^2} + \frac{2\mathbf{e}_2^\top \mathbf{K}_c \mathbf{e}_1 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \end{cases} \quad (3.6.7)$$

where $\mathbf{K}_c = \frac{1}{2}\overline{\mathbf{R}}\mathbf{K}\overline{\mathbf{R}}^\top$, \mathbf{K} is the principal curvature matrix of the surface,

$$\mathbf{K} = \begin{bmatrix} -\kappa_1 & 0 & 0 \\ 0 & -\kappa_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.6.8)$$

representing a canonical osculating paraboloid, and $\overline{\mathcal{R}}, \overline{\mathcal{T}}$ transforms from the local Monge surface patch coordinates $\overline{\mathbf{\Gamma}}$ to the camera coordinates $\mathbf{\Gamma}$ as $\mathbf{\Gamma}(t) = \overline{\mathcal{R}}(t)\overline{\mathbf{\Gamma}}(t) + \overline{\mathcal{T}}(t)$, so that $\overline{\mathcal{R}}(t) = \mathcal{R}(t)\overline{\mathcal{R}}_0$, and $\overline{\mathcal{T}}(t) = \mathcal{R}(t)\overline{\mathcal{T}}_0 + \mathcal{T}(t)$.

Proof. The second spatial derivatives of ρ are computed by considering an explicit osculating paraboloid equation as in the next section. We can then differentiate Equation (B.2.10) with respect to ξ, η and evaluate the result at the center point, to get:

$$\rho_{\xi\xi} = \frac{[4\rho\gamma^\top\mathbf{K}_ce_1 + 2\mathbf{N}_x]\rho_\xi + 2\gamma^\top\mathbf{K}_c\gamma\rho_\xi^2 + 2e_1^\top\mathbf{K}_ce_1\rho^2}{\mathbf{N} \cdot \gamma} \quad (3.6.9)$$

$$\rho_{\eta\eta} = \frac{[4\rho\gamma^\top\mathbf{K}_ce_2 + 2\mathbf{N}_y]\rho_\eta + 2\gamma^\top\mathbf{K}_c\gamma\rho_\eta^2 + 2e_2^\top\mathbf{K}_ce_2\rho^2}{\mathbf{N} \cdot \gamma} \quad (3.6.10)$$

$$\rho_{\xi\eta} = \frac{[2\rho\gamma^\top\mathbf{K}_ce_2 + \mathbf{N}_y]\rho_\xi + [2\rho\gamma^\top\mathbf{K}_ce_1 + \mathbf{N}_x]\rho_\eta + 2\gamma^\top\mathbf{K}_c\gamma\rho_\eta\rho_\xi + 2e_2^\top\mathbf{K}_ce_1\rho^2}{\mathbf{N} \cdot \gamma}, \quad (3.6.11)$$

where $\mathbf{K}_c = \frac{1}{2}\overline{\mathbf{R}}_0\mathbf{K}\overline{\mathbf{R}}_0^\top$ and ρ_ξ, ρ_η are given by Equation 3.6.3. Substituting the above values into (3.6.2), we get the desired second-order derivatives of flow. ■

3.7 Conclusion

In this chapter we presented the differential-geometric theory of projection and reconstruction of general curves from multiple views. We studied how the differential geometry of curves behaves under perspective projection, including the effects of intrinsic parameters. For instance, we studied how the tangent, curvature, and curvature derivative of a space curve projects onto an image, and how the motion of the camera and of the curve relate to the projections. We also gave formulas for reconstructing differential geometry, given differential geometry at corresponding points measured in two views. In particular, this gives a novel result of reconstructing space curve torsion, given corresponding points, tangents, curvatures, and curvature derivatives measured in two views. We determined that there are no constraints in two views – any pair of corresponding points with attributed tangents, curvatures, and curvature derivatives are possible matches, as long as the basic point epipolar constraint is satisfied. There is, however, a constraint in three views: from two views one can transfer differential geometry onto a third and enforce measurements to match the reprojection. This has been used in a recent practical work in curve-based multiview stereo by the authors [47].

This work is part of a greater effort of augmenting Multiple View Geometry to model curved structures. Work on practical camera calibration based on curves using the formulas in this chapter

is underway. We have also been studying the multiview differential geometry of surfaces and their shading.

Chapter 4

Camera Pose Estimation Using Curve Differential Geometry

4.1 Introduction

A key problem in the reconstruction of structure from multiple views of it is the determination of relative pose among cameras as well as the intrinsic parameters for each. The classical method is to place a “calibration jig”, *e.g.*, a set of checkerboard patterns, in the scene and to determine a set of corresponding points across views whose relative position is known. This correspondence can then be used to determine each camera’s intrinsic parameter matrix \mathcal{K}_{im} as well as the relative pose between pairs of cameras. While this is a well-known method, especially in photogrammetry, with well-crafted tools, *e.g.*, in MATLAB [15], the process is time consuming and labor intensive. It also leaves out a large number of images taken without a calibration jig in place.

A paradigm shift in the past couple of decades is the use of *interest points* or key features in object recognition and multiview geometry problems, which has led to the formulation of automatic methodologies to recover intrinsic parameters and relative pose. The paradigm shift came in response to a perceived failure on the part of segmentation techniques to produce reliable regions or contours that correspond to parts of objects in the image. Instead, the new paradigm relies on a set of isolated keypoints such as Harris corners [67] or SIFT/HOG [101] features which remain somewhat stable over view and other variations. As long as there is a sufficient number of keypoints between two views, a random selection of a few feature correspondences using RANSAC [54,69] can be verified by measuring the number of inlier features. These corresponding features then serve the same role as the keypoints selected from the calibration jig, although relative distances and angles in 3D are not available. Thus, these methods lead to an equivalence class of non-metric reconstructions. Constraints on the intrinsic parameters across the views and a single distance in 3D leads to a metric reconstruction [69].

These methods are currently in popular use through packages such as the Bundler used in

applications such as Phototourism [1]. The core technology underlying these methods are the notions of the *essential matrix* E , developed by Longuet-Higgins [99] where intrinsic parameters are available, and of the *fundamental matrix* F , devised by Faugeras, Luong and others [52, 102, 106] which works directly with image pixel coordinates thus not requiring intrinsic parameters. Typically a set of seven or eight pairs of corresponding points are used to obtain the fundamental matrix [69]. In general eight point algorithms work better in isolation, but when used with RANSAC seven point algorithms are preferred due to decreased probability of outliers and therefore reducing the number of necessary RANSAC iterations exponentially.

Two major drawbacks limit the applicability of automatic methods based on interest points. First, it is well-known that in practice the correlation of interest points works for views with a *limited baseline*, according to some estimates no greater than 30° [113], Figure 4.1(a). In contrast, certain curve fragments in the image, *e.g.*, those corresponding to sharp ridges, reflectance curves, *etc.*, persist stably over a much larger range of views. Second, the approach relies on an abundance of features so that some of them survive the various variations between views. While this is true in many scenes, as evidenced by the popularity of this approach, in numerous others this is not the case, such as (i) Homogeneous regions, *e.g.*, from man-made objects, corridors, *etc.*, Figure 4.1(b); (ii) Multiple moving objects require their own set of features which may not be sufficiently abundant without sufficient texture, Figure 4.1(c); (iii) Non-rigid objects require a rich set of features per roughly non-deforming patch, Figure 4.1(d). In all these cases, however, there is sufficient image curve structure, motivating augmenting the use of interest points with that of image curve structure.

The use of image curve fragments as the basic structure for auto-calibration is faced with two significant challenges. First, edge linking procedures do not produce curve segments which persist stably across images. Rather, an image curve fragment in one view may be segmented or grouped with other curve fragments. Thus, while the underlying curve geometry correlates well across views, the individual curve fragments do not, Figure 4.2(a-h). Second, even when the image curve fragments correspond exactly, there is an intra-curve correspondence ambiguity, Figure 4.2(i). This ambiguity prevents the use of corresponding curve points to solve for the unknown pose and intrinsic parameters. These challenges both point to the use of *small curve fragments*, as captured by the differential geometry of a curve at a point.

Related work: Previous work generally has relied on the concept of matching *epipolar tangencies* on *closed curves*, as reviewed briefly below: Let a 3D point $\mathbf{\Gamma}^w$, which has camera coordinates $\mathbf{\Gamma}^1$ and $\mathbf{\Gamma}^2$ in camera 1 and 2, respectively, have image projections γ^1 and γ^2 , respectively, *i.e.*, $\mathbf{\Gamma}^1 = \rho^1 \gamma^1$ and $\mathbf{\Gamma}^2 = \rho^2 \gamma^2$, where ρ^1 and ρ^2 are the depths of the point $\mathbf{\Gamma}^w$ in each camera, respectively. Let rotation matrix \mathcal{R} and translation vector \mathcal{T} describe the relative pose relating $\mathbf{\Gamma}^1$ and $\mathbf{\Gamma}^2$ as

$$\mathbf{\Gamma}^2 = R\mathbf{\Gamma}^1 + \mathcal{T} \quad (4.1.1)$$

Taking the cross-product with \mathcal{T} on both sides, gives

$$\mathcal{T} \times \mathbf{\Gamma}^2 = \mathcal{T} \times \mathcal{R}\mathbf{\Gamma}^1 \quad (4.1.2)$$

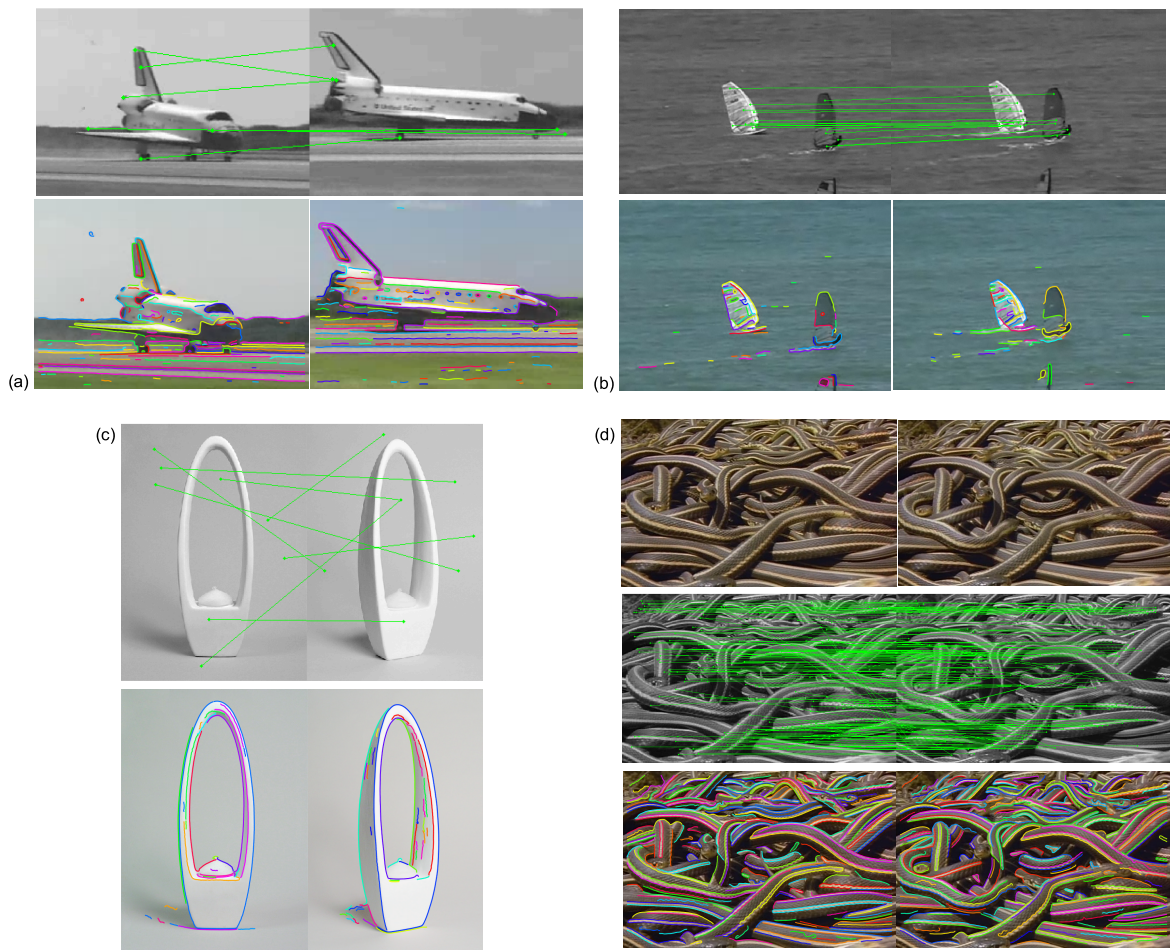


Figure 4.1: (a) Views with wide baseline separation may not have any interest points in common, but they often do share common curve structure. There may not always be sufficient interest points matching across views of homogenous objects such as for the sculpture in (c), but there is sufficient curve structure. (b) Each moving object requires its own set of features, but they may not be sufficient without a rich texture surface. (d) Non-rigid structures face the same issue.

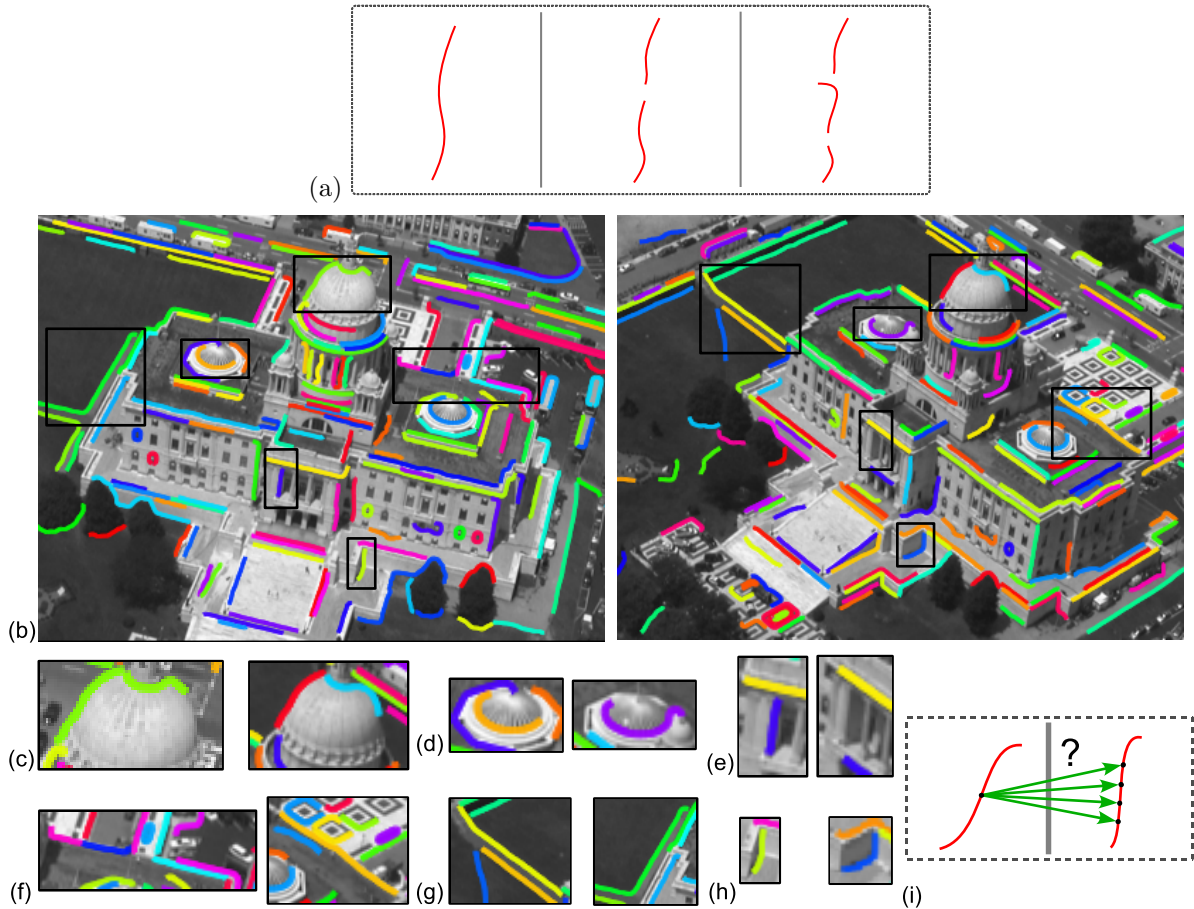


Figure 4.2: Challenges in using curve fragments in multiview geometry: (a) instabilities with slight changes in viewpoint, as shown for two views in (b) and zoomed in selectively in (c-h) real examples of edge grouping instabilities, such as a curve in one being broken into two in another view, a curve being linked onto background, a curve being detected in one view but absent in another, a curve being fragmented into various pieces at junctions in one view but fully linked in another view, different parts of a curve being occluded in different views, and a curve undergoing shape deformation from one view to the other. (i) Point correspondence ambiguity along the curve.

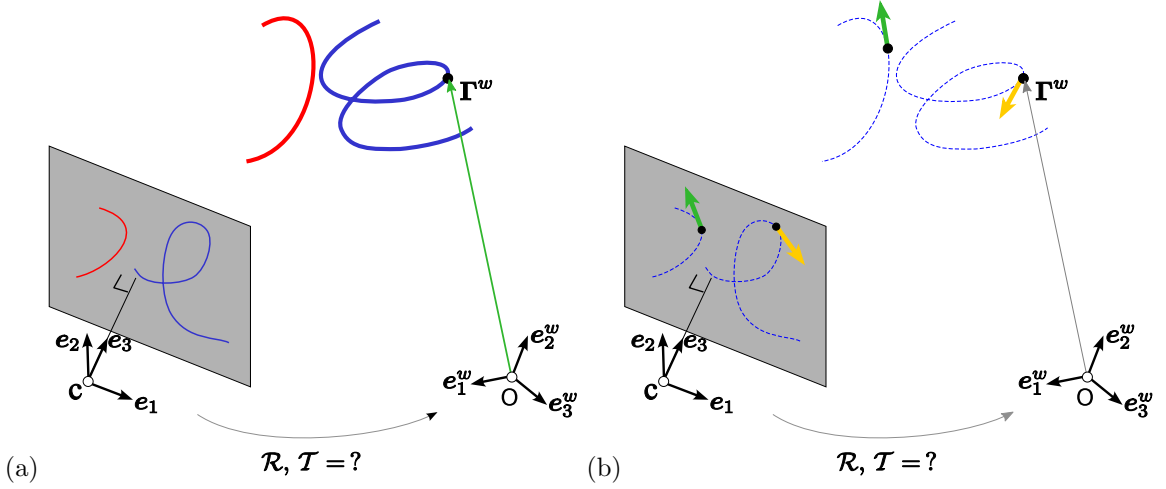


Figure 4.3: The problem of finding the camera pose \mathcal{R}, \mathcal{T} given space curves in a world coordinate system and their projections in an image coordinate system (a), and an approach to that consisting of (b) finding the camera pose \mathcal{R}, \mathcal{T} given 3D point-tangents (*i.e.*, local curve models) in a world coordinate system and their projections in an image coordinate system.

which after scalar product with $\mathbf{\Gamma}^2$ reduces to,

$$\mathbf{\Gamma}^2 \cdot (\mathcal{T} \times \mathcal{R}\mathbf{\Gamma}^1) = 0 \quad (4.1.3)$$

$$\mathbf{\Gamma}^{2\top} (\mathcal{T}_{\times} \mathcal{R}) \mathbf{\Gamma}^1 = 0, \quad (4.1.4)$$

where \mathcal{T}_{\times} is a skew-symmetric matrix formed from the translation vector \mathcal{T} , *i.e.*, $\mathcal{T}_{\times} \mathbf{\Gamma} = \mathcal{T} \times \mathbf{\Gamma}$. The essential matrix is then given by:

$$E = \mathcal{T}_{\times} \mathcal{R}. \quad (4.1.5)$$

Using $\mathbf{\Gamma}^1 = \rho^1 \gamma^1$ and $\mathbf{\Gamma}^2 = \rho^2 \gamma^2$ we get

$$\gamma^{2\top} E \gamma^1 = 0, \quad (4.1.6)$$

where E is the well-known *essential matrix* [99] which related the coordinates of two points in two views (with known intrinsic parameters). From a geometric perspective the camera center \mathbf{c}_1 of camera 1 which as the origin of the camera coordinate 1 maps to $\mathbf{\Gamma}^2 = 0 + \mathcal{T}$ according to Equation 4.1.1. Thus, \mathcal{T} is the vector from \mathbf{c}^1 to \mathbf{c}^2 . Second, the vector $E\gamma^1$ can be written as

$$E\gamma^1 = \mathcal{T}_{\times} \mathcal{R}\gamma^1 = \mathcal{T} \times \mathcal{R}\gamma^1, \quad (4.1.7)$$

and interpreted as follows: $\mathcal{R}\gamma^1$ is the vector from \mathbf{c}^1 to γ^1 in the coordinate of the second camera, Figure 4.4. The vector $\mathcal{T} \times \mathcal{R}\gamma^1$ is then orthogonal to the vector $\mathcal{T} = \overline{\mathbf{c}_1 \mathbf{c}_2}$ and $\mathcal{R}\gamma^1$, which span the epipolar plane. The essential constraint of Equation 4.1.6 implies $\gamma^2 \cdot (\mathcal{T} \times \mathcal{R}\gamma^1) = 0$, *i.e.*, that γ^2 lies on this epipolar plane. Thus, γ^2 is on the intersection of the epipolar plane and the second image

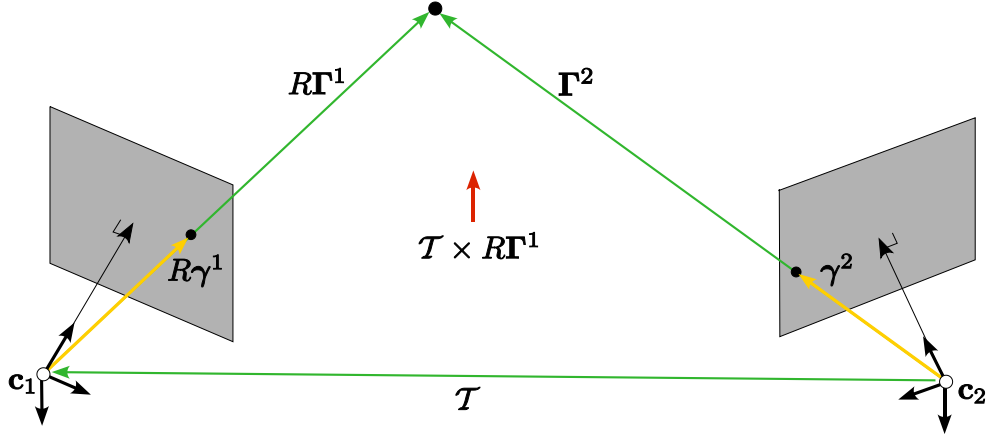


Figure 4.4: Interpretation of the essential matrix equation. All vectors are written with respect to the 3D coordinate system of camera 2.

plane, namely, the epipolar line. The same argument holds for γ^1 since the equation is symmetric, Figure 4.4. The epipolar line itself corresponding to γ^1 is the intersection of the epipolar plane and the image plane, *i.e.*,

$$\begin{cases} \gamma^{2\top} E \gamma_1 = 0 \\ e_3^\top \gamma^2 = 0. \end{cases} \quad (4.1.8)$$

The epipoles e_2^1 and e_1^2 , the intersections of $\overline{c_1 c_2}$ with the image 1 and 2, respectively, are the right and left null vectors of E , respectively, since they lie all on the epipolar planes, *i.e.*,

$$E e_2^1 = 0, \quad e_1^{2\top} E = 0. \quad (4.1.9)$$

Epipolar Tangency Constraint: The previous discussion of relations between two corresponding points can be extended to the relationship between the differential geometry of two curves $\gamma^1(s)$ in the first view and a curve $\gamma^2(s)$ in a second view, *i.e.*,

$$\gamma^{1\top}(s) E \gamma^2(s) = 0. \quad (4.1.10)$$

The tangents $t^1(s)$ and $t^2(s)$ are related by differentiating

$$g^1(s) t^{1\top}(s) E \gamma^2(s) + \gamma^{1\top}(s) E g^2(s) t^2(s) = 0, \quad (4.1.11)$$

where $g^1(s)$ and $g^2(s)$ are the respective speeds of parametrization of the curves $\gamma^1(s)$ and $\gamma^2(s)$. It is then clear that when one of the tangents $t^1(s)$ is along the epipolar plane also, *i.e.*, $t^{1\top}(s) E \gamma^2(s) = 0$ at a point s , then by necessity $\gamma^{1\top}(s) E t^2(s) = 0$. Thus, epipolar tangency in image 1 implies epipolar tangency in image 2 at the corresponding point, Figure 4.5.

Porrill and Pollard 1991: The epipolar tangency constraint was first shown in [126] who use linked edges and a coarsely specified epipolar geometry in the form of the essential matrix E . This initial estimate E is used to find a sparse set of epipolar tangencies, including those at corner in

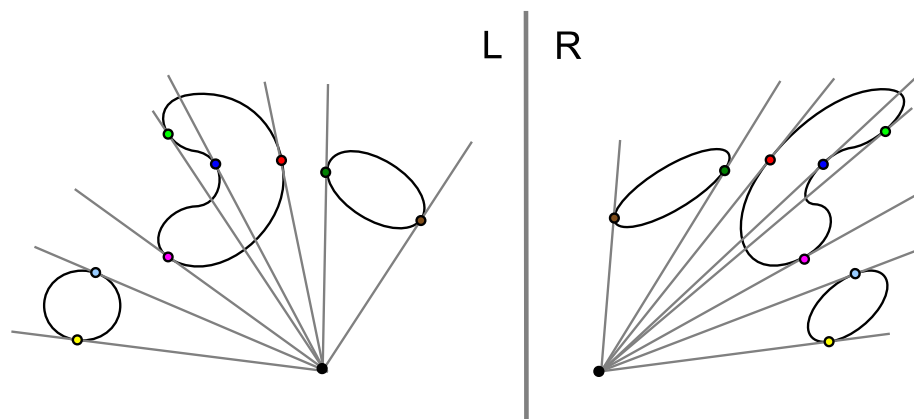


Figure 4.5: Correspondence of epipolar tangencies used in curve-based camera calibration. An epipolar line on the left, whose tangency at a curve is marked in a certain color, must correspond to the epipolar line on the right having tangency on the corresponding curve, marked with the same color. This concept works for both static curves and occluding contours.

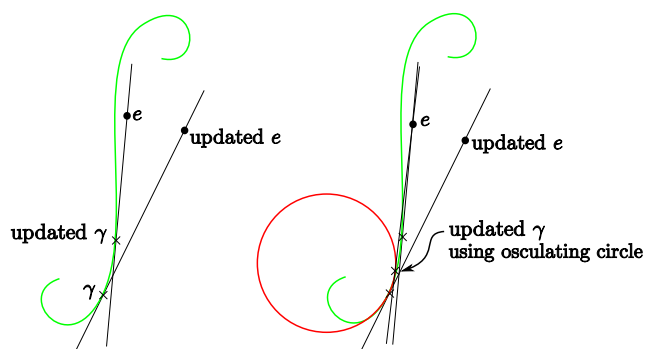


Figure 4.6: Illustrating the differential update of epipolar tangencies through the use of the osculating circle or curvature information.

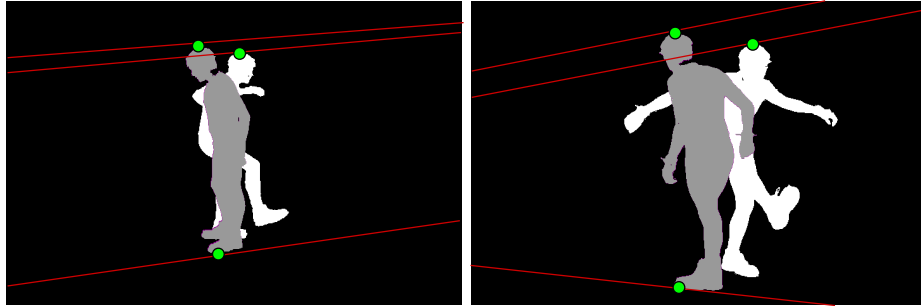


Figure 4.7: From [140]: given hypothesized epipoles, three corresponding epipolar lines are needed to solve for the complete epipolar geometry. The authors easily obtain two such correspondences at an instant $t = t_0$ (silhouette in gray) though the outermost epipolar tangencies, and a third correspondence is obtained by keeping the camera fixed but taking a snapshot at $t = t_1$ and using an outer epipolar tangency (at the white silhouette).

each view. They are matched from one view to another manually, which is then used to refine the estimate E , see Figure 4.6, by minimizing the residual $\gamma^{1\top}(s)E\gamma^2(s)$ over all matches in an iterative two-step scheme: the corresponding points are kept fixed and E is optimized in the first step in closed form and then E is kept fixed and the points are updated in a second step using a closed form solution based on an approximation of the curve as the osculating circle. This approach assumes that closed curves are available and that the order of epipolar tangencies are preserved.

Kahl and Heyden [85] consider the special case when four corresponding conics are available in two views, where the intrinsic parameters are unknown. In this algebraic approach, each pair of corresponding conics provide a pair of tangencies and therefore two constraints. Four pairs of conics are then needed. If the intrinsic parameters are available, then the absolute conic is known giving two constraints on the epipolar geometry, so that only 3 conic correspondences are required. This approach is only applied to synthetic data which shows the scheme to be extremely sensitive to even when a large number of conics (50) is used.

Kaminski and Shashua [86] extended this work to general algebraic curves viewed in multiple uncalibrated views. Specifically, they extend Kruppa's equations to describe the epipolar constraint of two projections of a general algebraic curve. The proposed solution requires solving systems of polynomial equations where recent computational algebraic geometry algorithms apply [31].

Sinha *et. al.* [140] consider a special configuration where multiple static cameras view a moving object with a controlled background, Figure 4.7. Observe that the epipolar geometry between any pair of cameras is fixed, each hypothesized pair of epipoles representing a point in 4D is then probed for a pair of epipolar tangencies across video frames. Specifically, two pairs of tangencies in one frame in time and a single pair of tangencies in another frame provide a constraint in that they must all intersect in the same point. This allows for an estimation of epipolar geometry for each pair of cameras, which are put together for refinement using bundle adjustment, providing intrinsic parameters and relative pose.

Overview. The paradigm explored in this chapter is that small curve fragments, or *curvelets*, can be used as the basic image structure to correlate across views. The intent is to use curve geometry as a complementary approach to the use of interest points in cases where these fail. Previous work in exploring local geometric groupings [144] has shown that tangent and curvature as well as the sign of curvature derivative can be reliably estimated. Unfortunately, differential geometry does not constrain point correspondence in two views and thus does not add any information in two views! However, it does significantly constrain three views or more. It is shown in [46] that the differential geometry at two corresponding points in two views reconstruct the differential geometry of the space curve they arise from. This constrains the differential geometry of corresponding curves in a third view. In the simplest case only first order differential geometry or tangent is used, and that is what this chapter explores.

The fundamental questions underlying the use of short curve fragments, essentially points augmented with differential-geometric attributes, is how many such curves are needed, what order of differential geometry is required, and how many views? This chapter explores the use of first order differential geometry with the minimum number of views, namely three. It considers two problems

Problem 1: For a camera with known intrinsic parameters, how many corresponding pairs of point-tangents in space specified in the world coordinates, and point-tangents in 2D specified in the image coordinates, are required to establish the pose of the camera with respect to the world coordinates, Figure 4.3(b)? This problem is useful when some reconstruction is already available either from other views [47] or because a 3D model of the object is available. It is also useful in the case of highly accurately calibrated moving binocular video to register the pose in one frame to that in another. In general, this is a basic Problem of interest in pose estimation, camera calibration, triangulation, *etc.*, in computer vision, robotics, computer graphics, photogrammetry and cartography.

Problem 2: For three cameras with known intrinsic parameters, how many corresponding triplets of point-tangents are needed to establish the relative pose of the cameras (*i.e.*, the trifocal relative pose)?

4.2 Determining Camera Pose from a Pair of 3D–2D Point-Tangent Correspondences

Our approach to treating Problem 1, in contrast to the global treatment as in algebraic geometry, is a local differential formulation: Given a local curve model consisting of points and tangents in 3D projecting to points and tangents in 2D, how many such correspondences are needed to estimate camera pose, Figure 4.3(b)? The classic result with correspondences of 3D points and 2D projections, is that 3 points are required to estimate camera pose [54]. Consider (2D, 3D) point correspondences, in the form of 3D points $(\Gamma_1, \Gamma_2, \dots, \Gamma_n)$ and 2D points $(\gamma_1, \gamma_2, \dots, \gamma_n)$, respectively. The coordinates of the 3D points are typically known from a calibration jig, or else from a previous reconstruction. This process is called *camera resectioning* in the photogrammetry literature (and by Hartley and Zisserman [69]), and also is known as *camera calibration* when this

is used with the purpose of obtaining the intrinsic parameter matrix \mathcal{K}_{im} where the camera pose relative to the calibration jig is not of interest. This is also related to the perspective n -point problem (PnP) originally introduced in [54], which can be stated as the recovery of the camera pose from n corresponding 3D-2D point pairs [77] or alternatively of depths [65].¹ The direct solution to P3P, also known as the *triangle pose problem*, is given in 1841 [63]. The direct solution equates the sides of the triangle with those of the vectors in the camera domain, *i.e.*,

$$\begin{cases} \|\rho_1 \gamma_1 - \rho_2 \gamma_2\|^2 = \|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|^2 = L_{12}^2 \\ \|\rho_2 \gamma_2 - \rho_3 \gamma_3\|^2 = \|\mathbf{\Gamma}_2^w - \mathbf{\Gamma}_3^w\|^2 = L_{23}^2 \\ \|\rho_3 \gamma_3 - \rho_1 \gamma_1\|^2 = \|\mathbf{\Gamma}_3^w - \mathbf{\Gamma}_1^w\|^2 = L_{31}^2 \end{cases} \quad (4.2.1)$$

where ρ_1, ρ_2, ρ_3 are depths in the camera coordinates. Expanding this equation gives a system of three quadratic equations

$$\begin{cases} \gamma_1^\top \gamma_1 \rho_1^2 - 2\gamma_1^\top \gamma_2 \rho_1 \rho_2 + \gamma_2^\top \gamma_2 \rho_2^2 = L_{12}^2 \\ \gamma_2^\top \gamma_2 \rho_2^2 - 2\gamma_2^\top \gamma_3 \rho_2 \rho_3 + \gamma_3^\top \gamma_3 \rho_3^2 = L_{23}^2 \\ \gamma_3^\top \gamma_3 \rho_3^2 - 2\gamma_3^\top \gamma_1 \rho_3 \rho_1 + \gamma_1^\top \gamma_1 \rho_1^2 = L_{31}^2 \end{cases} \quad (4.2.2)$$

Following traditional methods going back to the German mathematician Grunert in 1841 [63] and later Finsterwalder in 1937 [53], factor out one variable, say ρ .

$$\begin{cases} \rho_1^2 [\gamma_1^\top \gamma_1 - 2\gamma_1^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right) + \gamma_2^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right)^2] = L_{12}^2 \\ \rho_1^2 [\gamma_2^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right)^2 - 2\gamma_2^\top \gamma_3 \left(\frac{\rho_2}{\rho_1}\right) \left(\frac{\rho_3}{\rho_1}\right) + \gamma_3^\top \gamma_3 \left(\frac{\rho_3}{\rho_1}\right)^2] = L_{23}^2 \\ \rho_1^2 [\gamma_3^\top \gamma_3 \left(\frac{\rho_3}{\rho_1}\right)^2 - 2\gamma_3^\top \gamma_1 \left(\frac{\rho_3}{\rho_1}\right) + \gamma_1^\top \gamma_1] = L_{31}^2 \end{cases} \quad (4.2.3)$$

so that the system is reduced to two equations involving only the ratios $\frac{\rho_2}{\rho_1}$ and $\frac{\rho_3}{\rho_1}$,

$$\begin{cases} L_{23}^2 \left[\gamma_1^\top \gamma_1 - 2\gamma_1^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right) + \gamma_2^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right)^2 \right] = \\ L_{12}^2 \left[\gamma_2^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right)^2 - 2\gamma_2^\top \gamma_3 \left(\frac{\rho_2}{\rho_1}\right) \left(\frac{\rho_3}{\rho_1}\right) + \gamma_3^\top \gamma_3 \left(\frac{\rho_3}{\rho_1}\right)^2 \right] \\ L_{31}^2 \left[\gamma_1^\top \gamma_1 - 2\gamma_1^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right) + \gamma_2^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right)^2 \right] = \\ L_{12}^2 \left[\gamma_3^\top \gamma_3 \left(\frac{\rho_3}{\rho_1}\right)^2 - 2\gamma_3^\top \gamma_1 \left(\frac{\rho_3}{\rho_1}\right) + \gamma_1^\top \gamma_1 \right], \end{cases}$$

¹The camera resectioning problem does not assume that the camera projection matrix is fixed, while in the PnP problem the intrinsic parameters and camera pose remains fixed across views [78]

$$\left\{ \begin{array}{l} (L_{12}^2 - L_{23}^2)\gamma_2^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right)^2 + 2L_{23}^2\gamma_1^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right) - 2L_{12}^2\gamma_2^\top \gamma_3 \left(\frac{\rho_3}{\rho_1}\right) \left(\frac{\rho_2}{\rho_1}\right) \\ + L_{12}^2\gamma_3^\top \gamma_3 \left(\frac{\rho_3}{\rho_1}\right)^2 = L_{23}^2\gamma_1^\top \gamma_1 \\ L_{31}^2\gamma_2^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right)^2 - 2L_{31}^2\gamma_1^\top \gamma_2 \left(\frac{\rho_2}{\rho_1}\right) - L_{12}^2\gamma_3^\top \gamma_3 \left(\frac{\rho_3}{\rho_1}\right)^2 + 2L_{12}^2\gamma_3^\top \gamma_1 \left(\frac{\rho_3}{\rho_1}\right) \\ = (L_{12}^2 - L_{31}^2)\gamma_1^\top \gamma_1. \end{array} \right. \quad (4.2.4)$$

Grunert eliminates $\left(\frac{\rho_2}{\rho_1}\right)^2$ by a linear combination of these equations, which gives an expression for $\frac{\rho_2}{\rho_1}$ that is quadratic in $\frac{\rho_3}{\rho_1}$. After substituting this back into one of the above equations, a 4th order polynomial in $\frac{\rho_3}{\rho_1}$ results. Finsterwalder takes a slightly different approach that leads to a cubic polynomial by multiplying the second equation with λ and adding it to the first equations. The combined equation is quadratic in $\frac{\rho_2}{\rho_1}$ and $\frac{\rho_3}{\rho_1}$ and linear in λ . He solves for $\frac{\rho_3}{\rho_1}$ in terms of $\frac{\rho_2}{\rho_1}$ and λ using a standard solution to a quadratic which involves an expression of $\frac{\rho_2}{\rho_1}$ and λ under a square root. Since the equations hold for all λ , a λ for which this expression is a perfect square is sought. This involves finding λ as a root of a cubic polynomial, leading to an expression for $\frac{\rho_3}{\rho_1}$ that is linear in $\frac{\rho_2}{\rho_1}$, geometrically corresponding to two intersecting lines. This is better than Grunert's solution which involves solving a 4th order polynomial.

The camera resectioning problem can be solved using **three** 3D \leftrightarrow 2D point correspondences when the intrinsic parameters are known, and **six** points when the intrinsic parameters are not known. The camera pose can be solved using **four** point correspondences when only the focal length is unknown, but all the other intrinsic parameters are known [20], Table 4.1. We now show that when intrinsic parameters are known, **only a pair of point-tangent correspondences are required to estimate camera pose.**

Case	Unknowns	Min. # Corresp.
Calibrated (\mathcal{K}_{im} known)	Camera pose \mathcal{R}, \mathcal{T}	3
Focal length unknown	Pose \mathcal{R}, \mathcal{T} and f	4
Uncalibrated (\mathcal{K}_{im} unknown)	Camera model $\mathcal{K}_{im}, \mathcal{R}, \mathcal{T}$	6

Table 4.1: The number of 3D–2D point correspondences needed to solve for camera pose and intrinsic parameters.

Theorem 4.2.1. *Given a pair of 3D point tangents $\{(\mathbf{T}_1, \mathbf{T}_1), (\mathbf{T}_2, \mathbf{T}_2)\}$ described in a world coordinate system and their corresponding perspective projections $(\gamma_1, \mathbf{t}_1), (\gamma_2, \mathbf{t}_2)$, the pose of the camera \mathcal{R}, \mathcal{T} relative to the world coordinate system can be solved up to a finite number of solutions,*

assuming that the intrinsic parameters K_{im} are known, by solving for the system

$$\begin{cases} \gamma_1^\top \gamma_1 \rho_1^2 - 2\gamma_1^\top \gamma_2 \rho_1 \rho_2 + \gamma_2^\top \gamma_2 \rho_2^2 = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w), \\ Q(\rho_1, \rho_2) = 0, \end{cases} \quad (4.2.5)$$

where

$$\begin{aligned} Q(\rho_1, \rho_2) = & A^3(EH^2 - FHK + GK^2)^2 + AC^2(EJ^2 - FJL + GL^2)^2 \\ & - 2A^2C(EH^2 - FHK + GK^2)(EJ^2 - FJL + GL^2) + [-AB(EH^2 - FHK + GK^2) \\ & + BC(EJ^2 - FJL + GL^2)][A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)] \\ & + C[A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)]^2 = 0 \end{aligned} \quad (4.2.6)$$

is an eight degree polynomial, where the parameters A through L are defined as

$$\begin{cases} A = 1 - 2\gamma_1^\top \mathbf{t}_1 B_1 + \gamma_1^\top \gamma_1 B_1^2 \\ B = [2(\gamma_1^\top \mathbf{t}_1) - 2\gamma_1^\top \gamma_1 B_1] A_1 \\ C = (\gamma_1^\top \gamma_1) A_1^2 - 1 \\ E = 1 - 2\gamma_2^\top \mathbf{t}_2 B_2 + \gamma_2^\top \gamma_2 B_2^2 \\ F = [2(\gamma_2^\top \mathbf{t}_2) - 2\gamma_2^\top \gamma_2 B_2] A_2 \\ G = (\gamma_2^\top \gamma_2) A_2^2 - 1 \\ H = \gamma_1^\top \gamma_2 A_1 A_2 - (\mathbf{T}_1^w)^\top \mathbf{T}_2^w \\ J = [\gamma_2^\top \mathbf{t}_1 - \gamma_1^\top \gamma_2 B_1] A_2 \\ K = [\gamma_1^\top \mathbf{t}_2 - \gamma_1^\top \gamma_2 B_2] A_1 \\ L = \mathbf{t}_1^\top \mathbf{t}_2 - \gamma_2^\top \mathbf{t}_1 B_2 - \gamma_1^\top \mathbf{t}_2 B_1 + \gamma_1^\top \gamma_2 B_1 B_2, \end{cases} \quad (4.2.7)$$

where

$$\begin{cases} A_1 = \frac{(\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_1^w}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_1} \\ A_2 = \frac{(\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_2^w}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_2} \end{cases} \quad \begin{cases} B_1 = \frac{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \mathbf{t}_1}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_1} \\ B_2 = \frac{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \mathbf{t}_2}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_2}, \end{cases} \quad (4.2.8)$$

and where

$$\begin{cases} \rho_1 \frac{g_1}{G_1} = -\frac{A(EH^2 - FHK + GK^2) - C(EJ^2 - FJL + GL^2)}{A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)}, \\ \rho_2 \frac{g_2}{G_2} = -\frac{E(AH^2 - BHJ + CJ^2) - G(AK^2 - BKL + CL^2)}{E(2AHK - BHL - BKJ + 2CJL) - F(AK^2 - BKL + CL^2)}, \end{cases} \quad (4.2.9)$$

and

$$\begin{cases} \frac{\rho_1'}{G_1} = A_1 - B_1 \rho_1 \frac{g_1}{G_1} \\ \frac{\rho_2'}{G_2} = A_2 - B_2 \rho_2 \frac{g_2}{G_2}. \end{cases} \quad (4.2.10)$$

Finally,

$$\begin{cases} \mathcal{R} = [(\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w) \quad \mathbf{T}_1^w \quad \mathbf{T}_2^w]^{-1} [\rho_1 \gamma_1 - \rho_2 \gamma_2 \quad \rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1 \quad \rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_2] \\ \mathcal{T} = \rho_1 \gamma_1 - \mathcal{R} \mathbf{\Gamma}_1^w. \end{cases} \quad (4.2.11)$$

Proof. The proof proceeds by (i) writing all projection equations for each point and their derivatives in the simplest form involving \mathcal{R} , \mathcal{T} , depths ρ_1 and ρ_2 , depth derivatives ρ'_1 and ρ'_2 , and speed of parametrizations G_1 and G_2 , respectively; (ii) eliminating the translation \mathcal{T} by subtracting point equations; (iii) eliminating \mathcal{R} using dot products among equations, so that the remaining equations only have the local unknowns depths, depth derivatives, and parametrization speeds. (iv) eliminate the unknowns ρ'_1 and ρ'_2 , (v) eliminate g_1 and g_2 , (vi) solve for the remaining unknowns ρ_1 and ρ_2 , (vii) use these to compute all the unknowns.

The image point γ is related to the space point $\mathbf{\Gamma}$ through $\mathbf{\Gamma} = \rho \gamma$, where ρ is depth. The space point $\mathbf{\Gamma}$ in local coordinates is related to $\mathbf{\Gamma}^w$ in the world coordinates by a rotation matrix \mathcal{R} and translation \mathcal{T} through $\mathbf{\Gamma} = \mathcal{R} \mathbf{\Gamma}^w + \mathcal{T}$. Equating these at each of the two points gives

$$\begin{cases} \rho_1 \gamma_1 = \mathcal{R} \mathbf{\Gamma}_1^w + \mathcal{T} \\ \rho_2 \gamma_2 = \mathcal{R} \mathbf{\Gamma}_2^w + \mathcal{T}, \end{cases} \quad (4.2.12)$$

where ρ_1 and ρ_2 are the depth at image points γ_1 and γ_2 , respectively. By differentiating with respect to the parameters of γ_1 and γ_2 we have:

$$\begin{cases} \rho_1 g_1 \mathbf{t}_1 + \rho'_1 \gamma_1 = \mathcal{R} G_1 \mathbf{T}_1^w \\ \rho_2 g_2 \mathbf{t}_2 + \rho'_2 \gamma_2 = \mathcal{R} G_2 \mathbf{T}_2^w, \end{cases} \quad (4.2.13)$$

where ρ_1 and ρ_2 are depth derivatives with respect to the curve parameter, g_1 and g_2 are speeds of parametrization of γ_1 and γ_2 , respectively, and G_1 and G_2 are the speeds of parametrization of the space curves $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$, respectively. The vector Equations 4.2.12 and 4.2.13 represent 3 scalar equations for each point, so that there are 12 equations in all. The parametrization speeds g_1 and g_2 are arbitrary and can be set to 1 uniformly although we keep them in general form. The given quantities are γ , \mathbf{t} , and $\mathbf{\Gamma}^w$, \mathbf{T}^w at each point. The unknowns are \mathcal{R} , \mathcal{T} (6 unknowns), ρ , ρ' (4 unknowns), and the two speeds of the curve Γ at the two points, 12 unknowns in all. Therefore, in principle, two points should provide enough constraints to solve the problem.

First, \mathcal{T} is eliminated by subtracting the two Equations (4.2.12)

$$\rho_1 \gamma_1 - \rho_2 \gamma_2 = \mathcal{R} (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w), \quad (4.2.14)$$

which together with Equation 4.2.13 give a system of equations

$$\begin{cases} \rho_1 \gamma_1 - \rho_2 \gamma_2 = \mathcal{R} (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w) & (4.2.15) \\ \rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1 = \mathcal{R} \mathbf{T}_1^w & (4.2.16) \\ \rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_2 = \mathcal{R} \mathbf{T}_2^w. & (4.2.17) \end{cases}$$

At this stage, the unknowns are ρ_1 , ρ_2 , $\frac{\rho'_1}{G_1}$, $\frac{\rho'_2}{G_2}$, $\rho_1 \frac{g_1}{G_1}$, $\rho_2 \frac{g_2}{G_2}$, and \mathcal{R} , nine numbers in all, which can potentially be solved through the three vector equations (nine scalar equations) in (4.2.15)–(4.2.17). The number of unknowns can be reduced by eliminating \mathcal{R} in a second step. The matrix \mathcal{R} rotates three known vectors, $(\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)$, \mathbf{T}_1^w , and \mathbf{T}_2^w to the three unknown vectors on the left side of these equations, requiring a preservation of vector lengths and mutual angles. The length and relative angles are obtained from the known dot products, which do not involve \mathcal{R} at all. This provides six equations for the six unknowns $\{\rho_1, \rho_2, \frac{g_1}{G_1}, \frac{g_2}{G_2}, \frac{\rho'_1}{G_1}, \frac{\rho'_2}{G_2}\}$. Alternatively, we write these three equations in matrix form composed from the three vector equations (4.2.15)–(4.2.17), *i.e.*,

$$\begin{bmatrix} \rho_1 \gamma_1 - \rho_2 \gamma_2 & \rho \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1 & \rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_2 \end{bmatrix} = \mathcal{R} \begin{bmatrix} (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w) & \mathbf{T}_1^w & \mathbf{T}_2^w \end{bmatrix} \quad (4.2.18)$$

This is a system of six equations using product of the left hand matrix with its transpose which using $\mathcal{R}^\top \mathcal{R} = I$ gives

$$\left\{ \begin{array}{l} (\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top (\rho_1 \gamma_1 - \rho_2 \gamma_2) = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w) \\ (\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top (\rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1) = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_1^w \\ (\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top (\rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_1) = (\mathbf{\Gamma}_2^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_2^w \\ (\rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1)^\top (\rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1) = 1 \\ (\rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_2)^\top (\rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_2) = 1 \\ (\rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1)^\top (\rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_2) = (\mathbf{T}_1^w)^\top \mathbf{T}_2^w. \end{array} \right. \quad (4.2.19)$$

The first equation is a quadratic in ρ_1 and ρ_2

$$\gamma_1^\top \gamma_1 \rho_1^2 - 2\gamma_1^\top \gamma_2 \rho_1 \rho_2 + \gamma_2^\top \gamma_2 \rho_2^2 = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w), \quad (4.2.20)$$

which as a conic in the ρ_1 - ρ_2 plane with negative discriminant

$$(\gamma_1 \cdot \gamma_2)^2 - (\gamma_1 \cdot \gamma_1)(\gamma_2 \cdot \gamma_2) = -\|\gamma_1 \times \gamma_2\|^2 < 0 \quad (4.2.21)$$

is an ellipse. The ellipse is centered at the origin so we can check that it has real points by solving for ρ_1 when $\rho_2 = 0$, giving $\rho_1^2 \|\gamma_1\|^2 = \|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|^2$, or real roots $\rho_1 = \pm \frac{\|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|}{\|\gamma_1\|}$.

The remaining five equations involve the additional unknowns $\{\rho_1 \frac{g_1}{G_1}, \rho_2 \frac{g_2}{G_2}, \frac{\rho'_1}{G_1}, \frac{\rho'_2}{G_2}\}$. The latter appear in a linear form in the second and third equations, and in quadratic form in the last three equations. Thus, the terms $\frac{\rho'_1}{G_1}$ and $\frac{\rho'_2}{G_2}$ can be isolated from the second and third equations and then used in the last three equations

$$\left\{ \begin{array}{l} [(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_1] \frac{\rho'_1}{G_1} = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_1^w - [(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \mathbf{t}_1] \rho_1 \frac{g_1}{G_1} \\ [(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_2] \frac{\rho'_2}{G_2} = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_2^w - [(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \mathbf{t}_2] \rho_2 \frac{g_2}{G_2}, \end{array} \right. \quad (4.2.22)$$

or

$$\begin{cases} \frac{\rho'_1}{G_1} = \frac{(\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_1^w}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_1} - \left[\frac{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \mathbf{t}_1}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_1} \right] \rho_1 \frac{g_1}{G_1} = A_1 - B_1 \rho_1 \frac{g_1}{G_1} \\ \frac{\rho'_2}{G_2} = \frac{(\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_2^w}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_2} - \left[\frac{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \mathbf{t}_2}{(\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \gamma_2} \right] \rho_2 \frac{g_2}{G_2} = A_2 - B_2 \rho_2 \frac{g_2}{G_2}, \end{cases} \quad (4.2.23)$$

noting that A_1 , A_2 , B_1 , and B_2 depend on only two of the unknowns ρ_1 and ρ_2 . The last three equations in (4.2.19) can be expanded as

$$\begin{cases} \left(\rho_1 \frac{g_1}{G_1} \right)^2 + 2(\gamma_1^\top \mathbf{t}_1) \left(\rho_1 \frac{g_1}{G_1} \right) \left(\frac{\rho'_1}{G_1} \right) + (\gamma_1^\top \gamma_1) \left(\frac{\rho'_1}{G_1} \right)^2 = 1 \\ \left(\rho_2 \frac{g_2}{G_2} \right)^2 + 2(\gamma_2^\top \mathbf{t}_2) \left(\rho_2 \frac{g_2}{G_2} \right) \left(\frac{\rho'_2}{G_2} \right) + (\gamma_2^\top \gamma_2) \left(\frac{\rho'_2}{G_2} \right)^2 = 1 \\ (\mathbf{t}_1^\top \mathbf{t}_2) \left(\rho_1 \frac{g_1}{G_1} \right) \left(\rho_2 \frac{g_2}{G_2} \right) + (\gamma_2^\top \mathbf{t}_1) \left(\rho_1 \frac{g_1}{G_1} \right) \left(\frac{\rho'_2}{G_2} \right) + (\gamma_1^\top \mathbf{t}_2) \left(\rho_2 \frac{g_2}{G_2} \right) \left(\frac{\rho'_1}{G_1} \right) + \\ (\gamma_1^\top \gamma_2) \left(\frac{\rho'_1}{G_1} \right) \left(\frac{\rho'_2}{G_2} \right) = (\mathbf{T}_1^w)^\top \mathbf{T}_2^w. \end{cases}$$

Substituting $\frac{\rho'_1}{G_1}$ and $\frac{\rho'_2}{G_2}$ from Equations 4.2.23 gives

$$\begin{cases} \left(\rho_1 \frac{g_1}{G_1} \right)^2 + 2(\gamma_1^\top \mathbf{t}_1) \left(\rho_1 \frac{g_1}{G_1} \right) \left(A_1 - B_1 \left(\rho_1 \frac{g_1}{G_1} \right) \right) + (\gamma_1^\top \gamma_1) \left(A_1 - B_1 \left(\rho_1 \frac{g_1}{G_1} \right) \right)^2 = 1 \\ \left(\rho_2 \frac{g_2}{G_2} \right)^2 + 2(\gamma_2^\top \mathbf{t}_2) \left(\rho_2 \frac{g_2}{G_2} \right) \left(A_2 - B_2 \left(\rho_2 \frac{g_2}{G_2} \right) \right) + (\gamma_2^\top \gamma_2) \left(A_2 - B_2 \left(\rho_2 \frac{g_2}{G_2} \right) \right)^2 = 1 \\ (\mathbf{t}_1^\top \mathbf{t}_2) \left(\rho_1 \frac{g_1}{G_1} \right) \left(\rho_2 \frac{g_2}{G_2} \right) + (\gamma_2^\top \mathbf{t}_1) \left(\rho_1 \frac{g_1}{G_1} \right) \left(A_2 - B_2 \left(\rho_2 \frac{g_2}{G_2} \right) \right) + \\ (\gamma_1^\top \mathbf{t}_2) \left(\rho_2 \frac{g_2}{G_2} \right) \left(A_1 - B_1 \left(\rho_1 \frac{g_1}{G_1} \right) \right) + (\gamma_1^\top \gamma_2) \left(A_1 - B_1 \left(\rho_1 \frac{g_1}{G_1} \right) \right) \left(A_2 - B_2 \left(\rho_2 \frac{g_2}{G_2} \right) \right) \\ = (\mathbf{T}_1^w)^\top \mathbf{T}_2^w. \end{cases}$$

These three equations can be written in summary form using $x_1 = \rho_1 \frac{g_1}{G_1}$ and $x_2 = \rho_2 \frac{g_2}{G_2}$,

$$\begin{cases} Ax_1^2 + Bx_1 + C = 0 & (4.2.24) \\ Ex_2^2 + Fx_2 + G = 0 & (4.2.25) \\ H + Jx_1 + Kx_2 + Lx_1x_2 = 0, & (4.2.26) \end{cases}$$

and where A through L are only functions of the two unknowns ρ_1 and ρ_2 . Thus, the three Equations 4.2.24–4.2.26 after solving for x_1 and x_2 express a relationship between ρ_1 and ρ_2 , which together with Equation 4.2.20 can lead to a solution for ρ_1 and ρ_2 .

Equation 4.2.26, with given values for ρ_1 and ρ_2 , represents a rectangular hyperbola in the x_1 – x_2 plane, with asymptotes parallel to the axes, Figure 4.8. Moreover, each of the Equations 4.2.24 and 4.2.25 represents a pair of (real) lines in the same plane, parallel respectively to the x_2 and x_1 axes. We know that there is one mutual intersection of the aforementioned curves, but in general there should not be more than one.

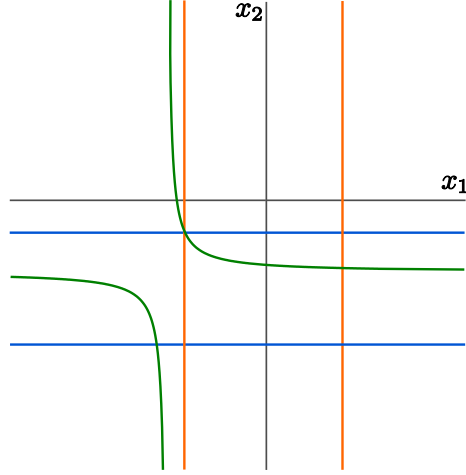


Figure 4.8: Diagram of the mutual intersection of Equations 4.2.24–4.2.26 in the x_1 - x_2 plane.

Specifically, the variables x_1 and x_2 can be solved by rewriting Equation 4.2.26 as

$$(H + Jx_1) + (K + Lx_1)x_2 = 0, \quad (4.2.27)$$

giving

$$x_2 = -\frac{H + Jx_1}{K + Lx_1}. \quad (4.2.28)$$

Using this expression in Equation 4.2.25 gives

$$E \frac{(H + Jx_1)^2}{(K + Lx_1)^2} - F \frac{H + Jx_1}{K + Lx_1} + G = 0, \quad (4.2.29)$$

or

$$E(H + Jx_1)^2 - F(H + Jx_1)(K + Lx_1) + G(K + Lx_1)^2 = 0. \quad (4.2.30)$$

Reorganizing as a quadratic in x_1 , this solves for x_1 which together with Equation 4.2.24 gives a constraint on the parameters depending on ρ_1 and ρ_2 ,

$$\begin{cases} (EJ^2 - FJL + GL^2)x_1^2 + (2EHJ - FHL - FJK + 2GKL)x_1 \\ \quad + (EH^2 - FHK + GK^2) = 0 \\ Ax_1^2 + Bx_1 + C = 0. \end{cases} \quad (4.2.31)$$

The quadratic term is eliminated by multiplying the first equation by A and the second equation by $(EJ^2 - FJL + GL^2)$ and subtracting, giving

$$\begin{aligned} [A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)]x_1 + \\ [A(EH^2 - FHK + GK^2) - C(EJ^2 - FJL + GL^2)] = 0, \end{aligned} \quad (4.2.33)$$

so that

$$x_1 = -\frac{A(EH^2 - FHK + GK^2) - C(EJ^2 - FJL + GL^2)}{A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)}. \quad (4.2.34)$$

Substituting back into Equation 4.2.32 gives

$$A \left[\frac{A(EH^2 - FHK + GK^2) - C(EJ^2 - FJL + GL^2)}{A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)} \right]^2 + \quad (4.2.35)$$

$$-B \frac{A(EH^2 - FHK + GK^2) - C(EJ^2 - FJL + GL^2)}{A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)} + C = 0,$$

or

$$\begin{aligned} & A^3(EH^2 - FHK + GK^2)^2 + AC^2(EJ^2 - FJL + GL^2)^2 \\ & - 2A^2C(EH^2 - FHK + GK^2)(EJ^2 - FJL + GL^2) + [-AB(EH^2 - FHK + GK^2) \\ & + BC(EJ^2 - FJL + GL^2)][A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)] \\ & + C[A(2EHJ - FHL - FJK + 2GKL) - B(EJ^2 - FJL + GL^2)]^2 = 0 \end{aligned} \quad (4.2.36)$$

The equation, after expressions for A, B, \dots, L are substituted in, gives a high order polynomial in ρ_1 and ρ_2 , *i.e.*, $Q(\rho_1, \rho_2) = 0$.² This equation together with Equation 4.2.20 represents a system of two equations in two unknowns

$$\begin{cases} \gamma_1^\top \gamma_1 \rho_1^2 - 2\gamma_1^\top \gamma_2 \rho_1 \rho_2 + \gamma_2^\top \gamma_2 \rho_2^2 = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w), \\ Q(\rho_1, \rho_2) = 0, \end{cases} \quad (4.2.37)$$

and gives a number of solutions for ρ_1 , and ρ_2 which in solve for the unknowns $\rho_1 \frac{g_1}{G_1}$, $\rho_2 \frac{g_2}{G_2}$, $\frac{\rho'_1}{G_1}$, and $\frac{\rho'_2}{G_2}$. Once these unknowns are solved for, the rotation \mathcal{R} can be obtained from the matrix equation (4.2.18). The translation \mathcal{T} is then solved from Equations 4.2.12 as

$$\mathcal{T} = \rho_1 \gamma_1 - \mathcal{R} \mathbf{\Gamma}_1^w. \quad (4.2.38)$$

■

Proposition 4.2.2. *The algebraic solutions to the systems (4.2.5) of Theorem 4.2.1 also require to satisfy imaging and other requirements enforced by the following inequalities*

$$\begin{cases} \rho_1 > 1, \rho_2 > 1 & (4.2.39) \\ \frac{g_1}{G_1} > 0, \frac{g_2}{G_2} > 0 & (4.2.40) \\ \det \mathcal{R} = \frac{\det \left[\rho_1 \gamma_1 - \rho_2 \gamma_2 \quad \rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho'_1}{G_1} \gamma_1 \quad \rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho'_2}{G_2} \gamma_2 \right]}{\det \left[\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w \quad \mathbf{T}_1^w \quad \mathbf{T}_2^w \right]} = H. & (4.2.41) \end{cases}$$

²This is an 8th order or less polynomial.

Proof. There are multiple solutions for ρ_1 and ρ_2 in Equation 4.2.37. Observe first that if $\rho_1, \rho_2, \mathcal{R}, \mathcal{T}$ are a solution, then so are $-\rho_1, -\rho_2, -\mathcal{R}$, and $-\mathcal{T}$, either by observing Equations 4.2.12 and 4.2.13. Only one of these two solutions are valid, however, as the camera geometry enforces positive depth, $\rho_1 > 0$ and $\rho_2 > 0$, so that solutions are sought only in the top right quadrant of the ρ_1 - ρ_2 space. In fact, the imaging geometry further restricts the points to lie in front of the camera so that $\rho_1 > 1$ and $\rho_2 > 2$.

Second, observe that the matrix \mathcal{R} can be a rotation matrix if it has determinant +1 or a reflection rotation matrix if it has determinant -1. This determinant can be computed for our solutions from Equations 4.2.18,

$$\det \mathcal{R} = \frac{\det \left[\rho_1 \gamma_1 - \rho_2 \gamma_2 \quad \rho_1 \frac{g_1}{G_1} \mathbf{t}_1 + \frac{\rho_1'}{G_1} \gamma_1 \quad \rho_2 \frac{g_2}{G_2} \mathbf{t}_2 + \frac{\rho_2'}{G_2} \gamma_2 \right]}{\det \left[\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w \quad \mathbf{T}_1^w \quad \mathbf{T}_2^w \right]}. \quad (4.2.42)$$

Among the solutions for ρ_1 and ρ_2 , we seek only those that give $\det \mathcal{R} > 0$. ■

The parametrization we have assumed in the space curve projects \mathbf{T} to the same half plane as \mathbf{t} in each view so that \mathbf{T} and \mathbf{t} need to point in the same direction, *i.e.*, $\mathbf{T} \cdot \mathbf{t} > 0$, or from Equations 4.2.16 and 4.2.17, $\frac{g_1}{G_1} > 0$ and $\frac{g_2}{G_2} > 0$.

Remark: One can express

$$\begin{cases} \mathbf{T}_1 = \cos \theta_1 \frac{\gamma_1}{\|\gamma_1\|} + \sin \theta_1 \mathbf{t}_1 \\ \mathbf{T}_2 = \cos \theta_2 \frac{\gamma_2}{\|\gamma_2\|} + \sin \theta_2 \mathbf{t}_2 \end{cases} \quad (4.2.43)$$

in which case the 4th and 5th equations above are automatically satisfied leading to 4 equations in 4 unknowns $\rho_1, \rho_2, \theta_1, \theta_2$.

$$\begin{cases} (\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top (\rho_1 \gamma_1 - \rho_2 \gamma_2) & = \|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|^2 \\ (\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \left(\cos \theta_1 \frac{\gamma_1}{\|\gamma_1\|} + \sin \theta_1 \mathbf{t}_1 \right) & = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_1^w \\ (\rho_1 \gamma_1 - \rho_2 \gamma_2)^\top \left(\cos \theta_2 \frac{\gamma_2}{\|\gamma_2\|} + \sin \theta_2 \mathbf{t}_2 \right) & = (\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w)^\top \mathbf{T}_2^w \end{cases} \quad (4.2.44)$$

This replaces polynomial equations with a trigonometric set of equations.

Practical Approach to Computing a Solution: Equations 4.2.37 can be viewed as the intersection of two curves in the $\rho_1 - \rho_2$ space. Since one of the curves to be intersected is an ellipse, it is possible to parametrize this curve by a bracketed parameter and then look for intersections with the curve of degree 8. This gives a higher order polynomial in a *single* unknown which is better than solving simultaneously the two equations of degree 2 and 8.

Proposition 4.2.3. *Solutions ρ_1 and ρ_2 to the quadratic Equation 4.2.20 can be parametrized as*

$$\begin{cases} \rho_1 = \frac{2\alpha t \cos \theta + \beta(1-t^2) \sin \theta}{1+t^2} \\ \rho_2 = \frac{-2\alpha t \sin \theta + \beta(1-t^2) \cos \theta}{1+t^2} \end{cases} \quad -1 \leq t \leq 1 \quad (4.2.45)$$

$$(4.2.46)$$

where

$$\tan(2\theta) = \frac{\gamma_1^\top \gamma_2}{\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2}, \quad 0 \leq 2\theta \leq \pi, \quad (4.2.47)$$

$$\alpha = \frac{\sqrt{2}\|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|}{\sqrt{(\gamma_1^\top \gamma_1 + \gamma_2^\top \gamma_2) + (\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2) \cos(2\theta) + 2\gamma_1^\top \gamma_2 \sin(2\theta)}}, \quad \alpha > 0, \quad (4.2.48)$$

$$\beta = \frac{\sqrt{2}\|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|}{\sqrt{(\gamma_1^\top \gamma_1 + \gamma_2^\top \gamma_2) - (\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2) \cos(2\theta) - 2\gamma_1^\top \gamma_2 \sin(2\theta)}}, \quad \beta > 0. \quad (4.2.49)$$

Proof. An ellipse centered at the origin with semi-axes of lengths $\alpha > 0$ and $\beta > 0$ and parallel to the coordinates satisfies

$$\left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2 = 1. \quad (4.2.50)$$

observing that $(2t)^2 + (1-t^2)^2 = (1+t^2)^2$, the coordinates x and y can be parametrized as

$$\begin{cases} x = \frac{2t}{1+t^2}\alpha \\ y = \frac{(1-t^2)}{1+t^2}\beta, \end{cases} \quad (4.2.51)$$

$$\begin{cases} x = \frac{2t}{1+t^2}\alpha \\ y = \frac{(1-t^2)}{1+t^2}\beta, \end{cases} \quad (4.2.52)$$

where $t \in (-\infty, \infty)$, with ellipse vertices identified at $-1, 0, 1$ and ∞ , as shown in Figure 4.9. For a general ellipse centered at the origin, the coordinates must be multiplied with the rotation matrix for angle θ , obtaining

$$\begin{cases} \rho_1 = \frac{2\alpha t \cos \theta + \beta(1-t^2) \sin \theta}{1+t^2} \\ \rho_2 = \frac{-2\alpha t \sin \theta + \beta(1-t^2) \cos \theta}{1+t^2}. \end{cases} \quad -1 \leq t \leq 1 \quad (4.2.53)$$

$$\begin{cases} \rho_1 = \frac{2\alpha t \cos \theta + \beta(1-t^2) \sin \theta}{1+t^2} \\ \rho_2 = \frac{-2\alpha t \sin \theta + \beta(1-t^2) \cos \theta}{1+t^2}. \end{cases} \quad -1 \leq t \leq 1 \quad (4.2.54)$$

Figure 4.9 illustrates this parametrization. Notice that the range of values of t which we need to consider certainly lies in the interval $[-1, 1]$ and in fact in a smaller interval where $\rho_1 > 0$ and $\rho_2 > 0$. Note that for t and $-\frac{1}{t}$ correspond to opposite points on the ellipse.

The above parametrization is for a general ellipse centered at the origin with α , β , and θ specified. The ellipse in Equation 4.2.20 is also centered at the origin. The parameters α , β , and θ for it can be then found by substitution of ρ_1 and ρ_2 in the parametric form in Equation 4.2.20. Specifically, writing

$$\begin{aligned} & \frac{\gamma_1^\top \gamma_1}{(1+t^2)^2} [4\alpha^2 t^2 \cos^2 \theta + \beta^2 (1-t^2)^2 \sin^2 \theta + 4\alpha\beta t (1-t^2) \sin \theta \cos \theta] + \\ & - \frac{2\gamma_1^\top \gamma_2}{(1+t^2)^2} [-4\alpha^2 t^2 \sin \theta \cos \theta + 2\alpha\beta t (1-t^2) \cos^2 \theta - 2\alpha\beta t (1-t^2) \sin^2 \theta] + \beta^2 (1-t^2)^2 \sin \theta \cos \theta \\ & - \frac{2\gamma_2^\top \gamma_2}{(1+t^2)^2} [4\alpha^2 t^2 \sin^2 \theta + \beta^2 (1-t^2)^2 \cos^2 \theta - 4\alpha\beta t (1-t^2) \sin \theta \cos \theta] = \|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|^2. \end{aligned} \quad (4.2.55)$$

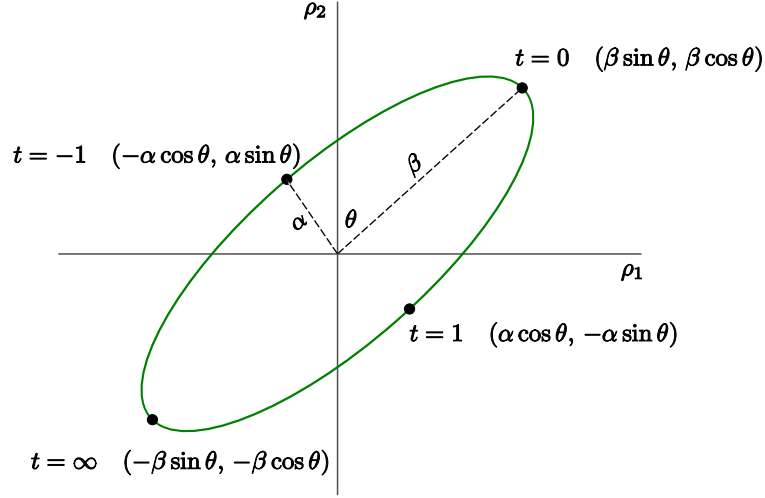


Figure 4.9: Diagram illustrating a parametrization of the ellipse by a parameter t .

Simplifying the equation as

$$\begin{aligned}
& [(\gamma_1^\top \gamma_1)4\alpha^2 t^2 - (\gamma_1^\top \gamma_2)4\alpha\beta t(1-t^2) + (\gamma_2^\top \gamma_2)\beta^2(1-t^2)^2] \cos^2 \theta + \\
& [(\gamma_1^\top \gamma_1)\beta^2(1-t^2)^2 + (\gamma_1^\top \gamma_2)4\alpha\beta t(1-t^2)(\gamma_2^\top \gamma_2)4\alpha^2 t^2] \sin^2 \theta + \\
& [(\gamma_1^\top \gamma_1)4\alpha\beta t(1-t^2) + (\gamma_1^\top \gamma_2)8\alpha^2 t^2 - (\gamma_1^\top \gamma_2)2\beta^2(1-t^2)^2 - (\gamma_2^\top \gamma_2)4\alpha\beta t(1-t^2)] \sin \theta \cos \theta \\
& = (1+t^2)^2 \|\Gamma_1^w - \Gamma_2^w\|^2
\end{aligned} \tag{4.2.56}$$

and using simple trigonometric identities $\cos^2 \theta = \frac{1+\cos(2\theta)}{2}$ and $\sin^2 \theta = \frac{1-\cos(2\theta)}{2}$, $\cos^2 \theta - \sin^2 \theta = \cos(2\theta)$ and $\sin(2\theta) = 2 \sin \theta \cos \theta$, this equation can be better simplified to

$$\begin{aligned}
& [(\gamma_1^\top \gamma_1)4\alpha^2 t^2 - (\gamma_1^\top \gamma_2)4\alpha\beta t(1-t^2) + (\gamma_2^\top \gamma_2)\beta^2(1-t^2)^2](1+\cos(2\theta)) + \\
& [(\gamma_1^\top \gamma_1)\beta^2(1-t^2)^2 + (\gamma_1^\top \gamma_2)4\alpha\beta t(1-t^2)(\gamma_2^\top \gamma_2)4\alpha^2 t^2](1-\cos(2\theta)) + \\
& [(\gamma_1^\top \gamma_1)4\alpha\beta t(1-t^2) + (\gamma_1^\top \gamma_2)8\alpha^2 t^2 - (\gamma_1^\top \gamma_2)2\beta^2(1-t^2)^2 - (\gamma_2^\top \gamma_2)4\alpha\beta t(1-t^2)] \sin(2\theta) \\
& = 2(1+t^2)^2 \|\Gamma_1^w - \Gamma_2^w\|^2.
\end{aligned} \tag{4.2.57}$$

which is an equation only involving the unknown θ ,

$$\begin{aligned}
& (\gamma_1^\top \gamma_1 + \gamma_2^\top \gamma_2)[4\alpha^2 t^2 + \beta^2(1-t^2)] + \\
& [(\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2)[4\alpha^2 t^2 - \beta^2(1-t^2)^2] - (\gamma_1^\top \gamma_2)8\alpha\beta t(1-t^2)] \cos(2\theta) \\
& [(\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2)4\alpha\beta t(1-t^2) + 2\gamma_1^\top \gamma_2[4\alpha^2 t^2 - \beta^2(1-t^2)^2]] \sin(2\theta) \\
& = 2(1+t^2)^2 \|\Gamma_1^w - \Gamma_2^w\|^2.
\end{aligned} \tag{4.2.58}$$

This equation holds for all values of t . $t = 0$,

$$(\gamma_1^\top \gamma_1 + \gamma_2^\top \gamma_2)\beta^2 - (\gamma_1^\top \gamma_2 - \gamma_2^\top \gamma_1)\beta^2 \cos(2\theta) - 2\gamma_1^\top \gamma_2 \beta^2 \sin(2\theta) = 2\|\Gamma_1^w - \Gamma_2^w\|, \tag{4.2.59}$$

giving

$$\beta^2 = \frac{2\|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|^2}{(\gamma_1^\top \gamma_1 + \gamma_2^\top \gamma_2) - (\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2) \cos(2\theta) - 2\gamma_1^\top \gamma_2 \sin(2\theta)}. \quad (4.2.60)$$

Similarly, at $t = 1$,

$$(\gamma_1^\top \gamma_1 + \gamma_2^\top \gamma_2)4\alpha^2 + (\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2)4\alpha^2 \cos(2\theta) + 2\gamma_1^\top \gamma_2 4\alpha^2 \sin(2\theta) = 8\|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|^2, \quad (4.2.61)$$

giving

$$\alpha^2 = \frac{2\|\mathbf{\Gamma}_1^w - \mathbf{\Gamma}_2^w\|^2}{(\gamma_1^\top \gamma_1 + \gamma_2^\top \gamma_2) + (\gamma_1^\top \gamma_1 - \gamma_2^\top \gamma_2) \cos(2\theta) + 2\gamma_1^\top \gamma_2 \sin(2\theta)}. \quad (4.2.62)$$

■

Remark: Note that $\tan \theta$ can be deduced from the value of $T := \tan(2\theta)$ by

$$\tan \theta = \begin{cases} \frac{\sqrt{1+T^2} - 1}{T} & \text{if } T > 0, \\ -\frac{\sqrt{1+T^2} - 1}{T} & \text{if } T < 0 \end{cases} \quad (4.2.63)$$

The result of substituting the parametrization of the ellipse into the other equation $Q = 0$, on clearing the denominators, appears to be an equation of degree 16 in t : The curve $Q = 0$ (degree 8) and the conic (degree 2) will meet in at most 16 real points. Note that t and $-\frac{1}{t}$ give diametrically opposite points on the ellipse, and the other curve $Q = 0$ is also symmetric with respect to the origin. So the polynomial $\tilde{Q}(t)$ which results from substituting for ρ_1 and ρ_2 in Q will be invariant to this substitution, and hence have the form:

$$\tilde{Q}(t) = q_0 + q_1 t + q_2 t^2 + \cdots + q_{16} t^{16}, \quad (4.2.64)$$

where $q_i = -q_{16-i}$ for i odd. At most 8 solutions will lie in the range $-1 < t \leq 1$, and indeed we are only interested in solutions which make $\rho_1 > 0$ and $\rho_2 > 0$.

Further considerations From careful examination of the polynomial P we find that:

- I) P is divisible by $\rho_1^4 \rho_2^4$. This is proved by multiple applications of the Hadamard lemma, namely by verifying that the first, second, and third partial derivatives of P with respect to ρ_1 , when substituting $\rho_1 = 0$, are identically zero, and similarly with ρ_2 .

- II) The residual polynomial

$$Q = \frac{P}{\rho_1^4 \rho_2^4} \quad (4.2.65)$$

has total degree 8 in the variables ρ_1 and ρ_2 .

- III) Since the original equations are all invariant when the signs of ρ_1 , ρ_2 , x_1 , x_2 are all reversed, which leaves the signs of ρ_1 and ρ_2 unchanged, the polynomial P has this property too:

$$P(-\rho_1, -\rho_2) = P(\rho_1, \rho_2). \quad (4.2.66)$$

The same is true of Equation 4.2.37.

4.3 Calibrating Using Three Point-Tangent Correspondences Across Three Views

The previous section concerns the case when two 3D point tangents $(\mathbf{\Gamma}_1, \mathbf{T}_1)$ and $(\mathbf{\Gamma}_2, \mathbf{T}_2)$ are available, *e.g.*, after reconstruction from two calibrated views, and addresses how these together with corresponding image point tangents (γ_1, \mathbf{t}_1) and (γ_2, \mathbf{t}_2) can be used to estimate camera pose. In general, when no calibrated views are available, however, the only information is the correspondence among image point-tangents, written in the form of (γ, \mathbf{t}) across several views. This case is the focus of this section.

Table 4.10 shows the number of point correspondences needed in two, three, and four views, with and without knowledge of intrinsic parameters, and, with and without knowledge of tangents at each point. In the case of points without knowledge of tangents, correspondence of 7 points in two views gives both pose and intrinsic parameters, although 8 points give the same though linear equations. In three or four views, 6 point correspondences are required but 7 points allow for linear equations. In four views, 6 points are required but these already give linear equations.

When the intrinsic parameters K_{im} are known, a general calculation due to Quan *et. al.* [129] gives a formula for the number of points necessary: each image point γ gives two constraints, so that P points in N views give $2NP$ equations. Each corresponding 3D point introduces 3 unknowns so that P points give $3P$ unknowns. Each camera introduces 6 unknowns so that N views correspond to $6(N - 1)$ unknowns relative to the first camera, which can be taken as the world coordinate system. An unknown needs to be subtracted because a scaling of the scene (all spatial distances, including depths relative to the camera as well as camera distance) does not change the observations. Thus, the set of observations are consistent with a one-parameter equivalence class of points and camera defined changes in the “scene scale”, an unknown which cannot be recovered from observed data. Therefore, the number of unknowns is reduced by one as only the equivalence class of solutions can be obtained! Thus, the total number of unknowns is $3P + 6(N - 1) - 1$. This implies that the number of constraints must exceed the number of unknowns to have finitely many solutions $2NP \geq 3P + 6N - 7$, or

$$P \geq 3 + \frac{2}{2N - 3}. \quad (4.3.1)$$

For two views, $N = 2$, $P \geq 5$, for three views $N = 3$, $P \geq 2\frac{2}{3}$, for four views $N = 4$, $P \geq 3\frac{2}{5}$, for five views $P \geq 3\frac{2}{7}$, *etc.* In other words, for three views or more at least four points are required, but these four points cannot be arbitrary as the situation is overconstrained.

The situation for point-tangents is similar: each point provides 2 constraints in position and 1 constraint in tangents, so that there are $3NP$ equations. But now there are 2 additional unknowns for each tangent in 3D, or $2P$ which in total gives $5P$ unknowns

$$3NP - 2P \geq 3P + 6N - 7 \quad (4.3.2)$$

or

$$P \geq 2 + \frac{3}{2N - 5}. \quad (4.3.3)$$

	Number of point correspondences required	Number of required oriented point correspondences	Degrees of freedom
Fundamental Matrix (Unknown K)	7 points nonlinear 8 points linear	7 points nonlinear 8 points linear	7
Essential Matrix (Known K)	5 points nonlinear 6 points linear	5 points nonlinear 6 points linear	5
Trifocal, unknown K	6 points nonlinear 7 points linear	Speculate 4	18
Trifocal, known K [Nister 2006]	4 overconstrained 3 underconstrained	Speculate 3	?
Quadrifocal, unknown K	6 linear equations		29
Quadrifocal, known K	??		??

# of views \ # of points	N = 2	N = 3	N = 4	N = 5	...	N = 100
Points	5	4	4	4	...	4
Oriented points	5	3	3	3	...	3
Oriented point + curvature	5	3	3	2	...	2
Oriented point, curvature, and curvature derivative	3	2	2	2	...	2

Figure 4.10: Variants of the multiview camera geometry estimation problem.

For $N = 2$ we have 5 points as before; for $N = 3$, $P \geq 2\frac{3}{4}$, for $N = 4$, $P \geq 2\frac{3}{7}$, for $N = 5$, $P \geq 2\frac{3}{10}$, so that for three views or more at least three points are required, and the equation is overconstrained.

The situation when points are endowed with both tangent and curvature is similar: points, tangents, and curvature gives 4 constraints per view for $4NP$ total. The unknowns for each 3D point is points (3), tangents (2), normals (1), and curvature (1), for a total of 7 per point, or $7P$, $(N - 2)P$ equations for curvatures, requiring

$$4NP \geq 4P + 6N - 7, \quad (4.3.4)$$

or

$$P \geq \frac{6N - 7}{4N - 7}. \quad (4.3.5)$$

For $N = 2$, $P \geq 5$, for $N = 3$, $P \geq 2\frac{1}{5}$, for $N = 4$, $P \geq 1\frac{8}{9}$, for $N = 5$, $P \geq 1\frac{10}{13}$, *etc.* Thus, for three views 3 points are required, but **for five views only two points are required!**

When points are endowed with curvature derivative, there is an additional constraint per point per view, *i.e.*, $5NP$. This introduces two additional 3D unknowns, *i.e.*, torsion and curvature derivative, for $9P$ unknowns, requiring $5NP \geq 9P + 6N - 7$ or

$$P \geq \frac{6N - 7}{5N - 9}. \quad (4.3.6)$$

For $N = 2$, $P \geq 5$, for $N = 3$, $P \geq 1\frac{5}{6}$, for $N = 4$, $P \geq 1\frac{6}{11}$, *etc.*, thus requiring only two points in three views.

Assume that three correspondences are given across three views, *i.e.*, that the point-tangents $(\gamma_1^1, \mathbf{t}_1^1)$, $(\gamma_2^1, \mathbf{t}_2^1)$, $(\gamma_3^1, \mathbf{t}_3^1)$ in the first view correspond to the three point-tangents $(\gamma_1^2, \mathbf{t}_1^2)$, $(\gamma_2^2, \mathbf{t}_2^2)$, $(\gamma_3^2, \mathbf{t}_3^2)$ in the second view, respectively and to the three points $(\gamma_1^3, \mathbf{t}_1^3)$, $(\gamma_2^3, \mathbf{t}_2^3)$, $(\gamma_3^3, \mathbf{t}_3^3)$ in the third view, respectively, and to the unknown 3D point-tangents $(\mathbf{\Gamma}_1^w, \mathbf{T}_1^w)$, $(\mathbf{\Gamma}_2^w, \mathbf{T}_2^w)$, and $(\mathbf{\Gamma}_3^w, \mathbf{T}_3^w)$, in space, respectively. Assume further that intrinsic parameters \mathcal{K}_{im}^1 , \mathcal{K}_{im}^2 , and \mathcal{K}_{im}^3 are given for all cameras. Our goal is to find the relative pose of cameras with respect to the world using $(\mathcal{R}_1, \mathcal{T}_1)$, $(\mathcal{R}_2, \mathcal{T}_2)$, $(\mathcal{R}_3, \mathcal{T}_3)$

$$\begin{cases} \mathbf{\Gamma}^1 = \mathcal{R}_1 \mathbf{\Gamma}^w + \mathcal{T}_1 \\ \mathbf{\Gamma}^2 = \mathcal{R}_2 \mathbf{\Gamma}^w + \mathcal{T}_2 \\ \mathbf{\Gamma}^3 = \mathcal{R}_3 \mathbf{\Gamma}^w + \mathcal{T}_3 \end{cases} \quad (4.3.7)$$

as well as relative pose among cameras $(\mathcal{R}_{21}, \mathcal{T}_{21})$, $(\mathcal{R}_{31}, \mathcal{T}_{31})$, $(\mathcal{R}_{32}, \mathcal{T}_{32})$

$$\begin{cases} \mathbf{\Gamma}^2 = \mathcal{R}_{21} \mathbf{\Gamma}^1 + \mathcal{T}_{21} \\ \mathbf{\Gamma}^3 = \mathcal{R}_{31} \mathbf{\Gamma}^1 + \mathcal{T}_{31} \\ \mathbf{\Gamma}^3 = \mathcal{R}_{32} \mathbf{\Gamma}^2 + \mathcal{T}_{32} \end{cases} \quad (4.3.8)$$

Note that

$$\mathbf{\Gamma}^2 = \mathcal{R}_{21}(\mathcal{R}_1 \mathbf{\Gamma}^w + \mathcal{T}_1) + \mathcal{T}_{21} = \mathcal{R}_{21} \mathcal{R}_1 \mathbf{\Gamma}^w + \mathcal{R}_{21} \mathcal{T}_1 + \mathcal{T}_{21}, \quad (4.3.9)$$

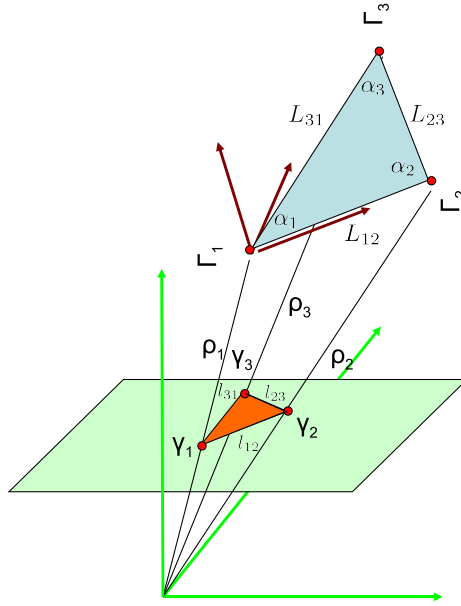


Figure 4.11: Intrinsic coordinate system is in brown, while the view coordinate system is in green.

so that $\mathcal{R}_2 = \mathcal{R}_{21}\mathcal{R}_1$ and $\mathcal{T}_2 = \mathcal{R}_{21}\mathcal{T}_1 + \mathcal{T}_{21}$ and

$$\begin{cases} \mathcal{R}_{21} = \mathcal{R}_2\mathcal{R}_1^\top \\ \mathcal{R}_{31} = \mathcal{R}_3\mathcal{R}_1^\top \\ \mathcal{R}_{32} = \mathcal{R}_3\mathcal{R}_2^\top \end{cases} \quad \begin{cases} \mathcal{T}_{21} = -\mathcal{R}_{21}\mathcal{T}_1 + \mathcal{T}_2 \\ \mathcal{T}_{31} = -\mathcal{R}_{31}\mathcal{T}_1 + \mathcal{T}_3 \\ \mathcal{T}_{32} = -\mathcal{R}_{32}\mathcal{T}_2 + \mathcal{T}_3. \end{cases} \quad (4.3.10)$$

We also have

$$\begin{cases} \mathbf{\Gamma}_1^1 = \rho_1^1\gamma_1^1 \\ \mathbf{\Gamma}_2^1 = \rho_3^1\gamma_2^1 \\ \mathbf{\Gamma}_3^1 = \rho_3^1\gamma_3^1 \end{cases} \quad \begin{cases} \mathbf{\Gamma}_1^2 = \rho_1^2\gamma_1^2 \\ \mathbf{\Gamma}_2^2 = \rho_3^2\gamma_2^2 \\ \mathbf{\Gamma}_3^2 = \rho_3^2\gamma_3^2 \end{cases} \quad \begin{cases} \mathbf{\Gamma}_1^3 = \rho_1^3\gamma_1^3 \\ \mathbf{\Gamma}_2^3 = \rho_3^3\gamma_2^3 \\ \mathbf{\Gamma}_3^3 = \rho_3^3\gamma_3^3. \end{cases} \quad (4.3.11)$$

Intuitively, assuming that $(\mathcal{R}_{12}, \mathcal{T}_{12})$ is known, the 3D point-tangents $(\mathbf{\Gamma}_1, \mathbf{T}_1)$, $(\mathbf{\Gamma}_2, \mathbf{T}_2)$, $(\mathbf{\Gamma}_3, \mathbf{T}_3)$ can be reconstructed from views 1 and 2. Then, applying Problem 1 to a pair of these, say $(\mathbf{\Gamma}_1, \mathbf{T}_1)$ and $(\mathbf{\Gamma}_2, \mathbf{T}_2)$ and the corresponding (γ_1^3, t_1^3) and (γ_2^3, t_2^3) in the third view, we can determine $(\mathcal{R}_3, \mathcal{T}_3)$. Repeating this with a different pair, say $(\mathbf{\Gamma}_1, \mathbf{T}_1)$ and $(\mathbf{\Gamma}_3, \mathbf{T}_3)$ gives a different estimate for $(\mathcal{R}_3, \mathcal{T}_3)$. Equating these six equations should give \mathcal{R}, \mathcal{T} . This reconstruction is then used in determining $(\mathcal{R}_3, \mathcal{T}_3)$.

Alternatively, the three unknown $(\rho_1^1, \rho_2^1, \rho_3^1)$ from the first view, Figure 4.11, determine $(\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \mathbf{\Gamma}_3)$, and the three unknowns $(\rho_1', \rho_2', \rho_3')$ determine the tangents $(\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3)$. These six unknowns, like the six unknowns in $(\mathcal{R}, \mathcal{T})$ above can be constrained and solved. Both these options, however, favor one view as special and are not symmetric, and also involve difficult nonlinear rotations and projections. Instead, we choose a more geometric set of unknowns. Specifically, let the lengths of

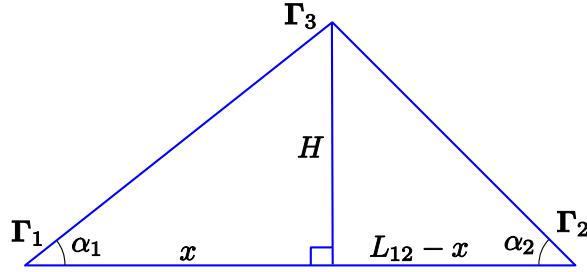


Figure 4.12:

the triangle connecting Γ_1 , Γ_2 , and Γ_3 be

$$\begin{cases} L_{12} = \|\Gamma_1 - \Gamma_2\| \\ L_{23} = \|\Gamma_2 - \Gamma_3\| \\ L_{31} = \|\Gamma_3 - \Gamma_1\|, \end{cases} \quad (4.3.12)$$

as shown in Figure 4.11. For the tangents we use the two angles T_1 , T_2 , and T_3 make with the sides T_1 with $\Gamma_1\Gamma_2$ and $\Gamma_1\Gamma_3$, θ_{12} , θ_{13} , respectively, and T_3 with Γ_{31} , Γ_{32} , θ_{31} , θ_{32} , respectively. The conjecture is that these geometric quantities are symmetric with respect to views and involve simple relations.

First, consider that the intra distance of the three non-collinear points Γ_1 , Γ_2 , and Γ_3 are given as L_{12} , L_{23} , L_{31} as in Equation 4.3.12, but not the coordinates of Γ_1 , Γ_2 , and Γ_3 . Then we can set a local coordinate system where Γ_1 is the origin, $\Gamma_1\Gamma_2$ defines the first axis, $\Gamma_1\Gamma_2 \times \Gamma_1\Gamma_3$ defines the third axis, and $\Gamma_1\Gamma_2 \times (\Gamma_1\Gamma_2 \times \Gamma_1\Gamma_3)$ defines the second axis. We can now use the following proposition to find \mathcal{R} and \mathcal{T} .

We take L_{12} , α_1 , and α_2 to fully describe the geometry of the triangle, modulo a rigid transformation. Then

$$\begin{cases} \Gamma_1^w = (0, 0, 0)^\top \\ \Gamma_2^w = (L_{12}, 0, 0)^\top \\ \Gamma_3^w = (L_{13} \cos \alpha, L_{13} \sin \alpha, 0)^\top, \end{cases} \quad (4.3.13)$$

where L_{13} can be obtained from solving

$$\begin{cases} \tan \alpha_1 = \frac{H}{x} \\ \tan \alpha_2 = \frac{H}{L_{12} - x} \end{cases} \implies \frac{\tan \alpha_1}{\tan \alpha_2} = \frac{L_{12} - x}{x} \implies \frac{\tan \alpha_1}{\tan \alpha_2} + 1 = \frac{L_{12}}{x} \quad (4.3.14)$$

giving

$$x = \frac{L_{12}}{\frac{\tan \alpha_1}{\tan \alpha_2} + 1} \quad (4.3.15)$$

so that

$$\begin{cases} L_{31} = \frac{x}{\cos \alpha_1} = \frac{L_{12}}{\cos \alpha_1 \left(\frac{\tan \alpha_1}{\tan \alpha_2} + 1 \right)} \\ L_{23} = \frac{L_{12} - x}{\cos \alpha_2} = \frac{\frac{\tan \alpha_1}{\tan \alpha_2} L_{12}}{\cos \alpha_2 \left(\frac{\tan \alpha_1}{\tan \alpha_2} + 1 \right)} \end{cases} \quad (4.3.16)$$

The angle $\alpha_3 = \pi - (\alpha_1 + \alpha_2)$.

Proposition 4.3.1. *Suppose three non-collinear points $\{\Gamma_1, \Gamma_2, \Gamma_3\}$ in the camera coordinate system have world coordinate representation $\{\Gamma_1^w, \Gamma_2^w, \Gamma_3^w\}$. Then the two coordinate systems are related by*

$$\Gamma = \mathcal{R}\Gamma^w + \mathcal{T}, \quad (4.3.17)$$

where

$$\mathcal{R} = \begin{bmatrix} \Gamma_2 - \Gamma_1 & \Gamma_3 - \Gamma_1 & (\Gamma_2 - \Gamma_1) \times (\Gamma_3 - \Gamma_1) \\ \Gamma_2^w - \Gamma_1^w & \Gamma_3^w - \Gamma_1^w & (\Gamma_2^w - \Gamma_1^w) \times (\Gamma_3^w - \Gamma_1^w) \end{bmatrix}^{-1} \quad (4.3.18)$$

and $\mathcal{T} = \Gamma_1 - \mathcal{R}\Gamma_1^w$. In other words, three corresponding pairs of points fully determine camera pose.

Proof. The translation is easily eliminated from the system of equations

$$\begin{cases} \Gamma_1 = \mathcal{R}\Gamma_1^w + \mathcal{T} \\ \Gamma_2 = \mathcal{R}\Gamma_2^w + \mathcal{T} \\ \Gamma_3 = \mathcal{R}\Gamma_3^w + \mathcal{T} \end{cases} \quad (4.3.19)$$

by pairwise subtraction giving

$$\begin{cases} \Gamma_1 - \Gamma_2 = \mathcal{R}(\Gamma_1^w - \Gamma_2^w) \\ \Gamma_1 - \Gamma_3 = \mathcal{R}(\Gamma_1^w - \Gamma_3^w). \end{cases} \quad (4.3.20)$$

Observe that the vector $(\Gamma_1 - \Gamma_2) \times (\Gamma_1 - \Gamma_3)$ cannot be written as a linear sum of $\Gamma_1 - \Gamma_2$ and $\Gamma_1 - \Gamma_3$. We have obtained by observing that

$$(\Gamma_1 - \Gamma_3) \times (\Gamma_2 - \Gamma_3) = \mathcal{R}[(\Gamma_1^w - \Gamma_2^w) \times (\Gamma_1^w - \Gamma_3^w)] \quad (4.3.21)$$

so that the matrix formed from the three vectors $\Gamma_1 - \Gamma_3, \Gamma_2 - \Gamma_3$ and $(\Gamma_1 - \Gamma_3) \times (\Gamma_2 - \Gamma_3)$ is \mathcal{R} times that in the world coordinates,

$$\begin{bmatrix} \Gamma_1 - \Gamma_2 & \Gamma_1 - \Gamma_3 & (\Gamma_1 - \Gamma_2) \times (\Gamma_1 - \Gamma_3) \end{bmatrix} = \mathcal{R} \begin{bmatrix} \Gamma_1^w - \Gamma_2^w & \Gamma_1^w - \Gamma_3^w & (\Gamma_1^w - \Gamma_2^w) \times (\Gamma_1^w - \Gamma_3^w) \end{bmatrix}, \quad (4.3.22)$$

and this matrix equation can be solved to find \mathcal{R} . \mathcal{T} can be found from the first equation in (4.3.19). ■

Corollary 4.3.2. *In the case when $\Gamma_1^{w\top} = (0, 0, 0)$, $\Gamma_2^{w\top} = (L_{12}, 0, 0)$, and $\Gamma_3^{w\top} = (L_{13} \cos \alpha_1, L_{13} \sin \alpha_1, 0)$ we have*

$$(\Gamma_2^w - \Gamma_1^w) = (L_{12}, 0, 0) \quad (4.3.23)$$

$$(\Gamma_3^w - \Gamma_1^w) = (L_{13} \cos \alpha_1, L_{13} \sin \alpha_1, 0) \quad (4.3.24)$$

and

$$(\mathbf{\Gamma}_2^w - \mathbf{\Gamma}_1^w) \times (\mathbf{\Gamma}_3^w - \mathbf{\Gamma}_1^w) = (0, 0, L_{13}L_{12} \sin \alpha_1)^\top, \quad (4.3.25)$$

so that

$$\begin{bmatrix} \mathbf{\Gamma}_2^w - \mathbf{\Gamma}_1^w & \mathbf{\Gamma}_3^w - \mathbf{\Gamma}_1^w & (\mathbf{\Gamma}_2^w - \mathbf{\Gamma}_1^w) \times (\mathbf{\Gamma}_3^w - \mathbf{\Gamma}_1^w) \end{bmatrix} = \begin{bmatrix} L_{12} & L_{13} \cos \alpha_1 & 0 \\ 0 & L_{13} \sin \alpha_1 & 0 \\ 0 & 0 & L_{13}L_{12} \sin \alpha_1 \end{bmatrix} \quad (4.3.26)$$

which from Equation 4.3.22 gives

$$\mathcal{R} = \begin{bmatrix} \mathbf{\Gamma}_1 - \mathbf{\Gamma}_3 & \mathbf{\Gamma}_2 - \mathbf{\Gamma}_3 & (\mathbf{\Gamma}_1 - \mathbf{\Gamma}_3) \times (\mathbf{\Gamma}_2 - \mathbf{\Gamma}_3) \end{bmatrix} \begin{bmatrix} \frac{1}{L_{12}} & \frac{-1 \cos \alpha_1}{L_{12} \sin \alpha_1} & 0 \\ 0 & \frac{1}{L_{13} \sin \alpha_1} & 0 \\ 0 & 0 & \frac{1}{L_{13}L_{12} \sin \alpha_1} \end{bmatrix} \quad (4.3.27)$$

or

$$\mathcal{R} = \begin{bmatrix} n\rho_1\gamma_1 - \rho_3\gamma_3 & \rho_2\gamma_2 - \rho_3\gamma_3 & \rho_1\rho_2\gamma_1 \times \gamma_2 - \rho_1\rho_3\gamma_1 \times \gamma_3 - \rho_3^2\gamma_3 \times \gamma_2 \end{bmatrix} \begin{bmatrix} \frac{1}{L_{12}} & \frac{-1 \cos \alpha_1}{L_{12} \sin \alpha_1} & 0 \\ 0 & \frac{1}{L_{13} \sin \alpha_1} & 0 \\ 0 & 0 & \frac{1}{L_{13}L_{12} \sin \alpha_1} \end{bmatrix} \quad (4.3.28)$$

4.4 Future Directions

The short-term research direction would be to develop a more explicit numerical scheme for implementing Theorem 4.2.1. The ellipse could be parametrized by rational functions in terms of a parameter in the $[-1, 1]$ range, and we could find roots of the resulting single-variable polynomial in this finite interval.

We have also been working on the problem of determining trinocular relative pose from corresponding point-tangents across 3 views. This would allow for complete curve-based structure from motion systems starting from a set of images without any initial calibration. These systems would start by building an initial solution for 3 views, refine the solution using bundle adjustment, then use our Theorem 4.2.1 to add new views to the system one by one.

Chapter 5

3D Curve Sketch: Curve-Based Stereo Reconstruction and Calibration

5.1 Introduction

The 3D reconstruction of scenes from images taken from multiple cameras and the calibration of these cameras are fundamental problems in Computer Vision. The state-of-the-art approaches to these problems either find correspondences among *isolated interest points* which give a 3D point cloud reconstruction of the scene, or are *intensity-based multiview stereo methods*, which give detailed mesh reconstructions of simple objects but use controlled acquisition. These approaches have been successful for select domains of application, resulting in autocalibration and useful 3D reconstructions. However, their requirements/assumptions are not applicable in general, motivating a novel approach based on image curve content to complement existing approaches.

The application of **interest-point-based methods** have been successful in scenes with texture-rich images, such as in Phototourism [1, 125]. Despite their success, these methods are not applicable in general settings. First, they assume an abundance of interest points per independently moving object, possibly only with a fair degree of texture in projected images. Such a rich texture is not always available in scenes with homogeneous regions, *e.g.*, some man-made environments, or when objects project to a small number of pixels, relative to the basis of computation of interest points. In these cases, there may be sufficient image curve geometry, however, to reconstruct the scene and recover the cameras.

Second, the stability of interest points is significantly reduced as the baseline exceeds 30° [113], so these methods are fundamentally restricted to a limited range of baselines. In contrast, certain curve features, such as sharp ridges on a building, persist over a much greater range of views. Third,

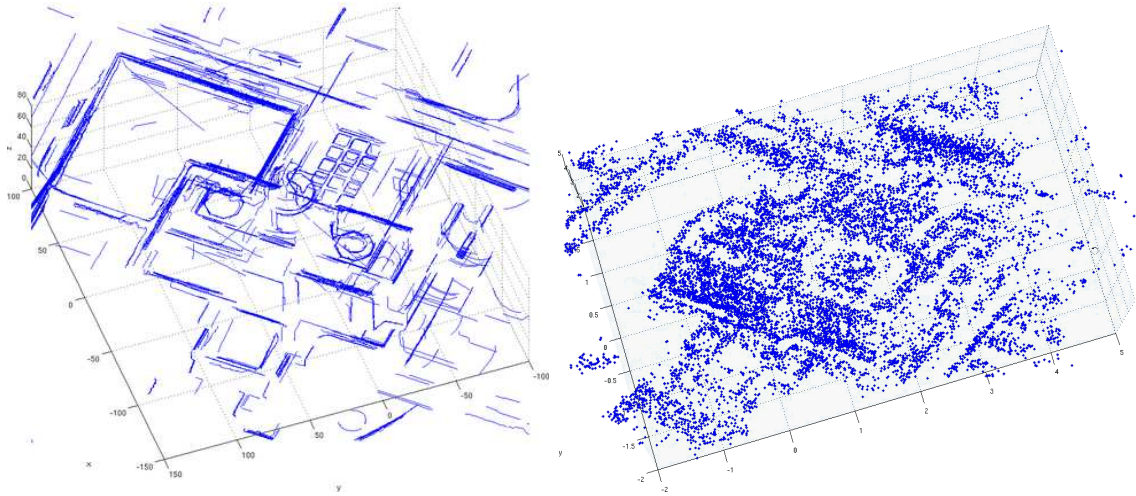


Figure 5.1: The 3D reconstruction as a 3D curve sketch (left) gives explicit geometry in contrast to the 3D cloud of point reconstruction (right). In essence, the proposed method produces a cloud of 3D curve fragments.

reconstruction using interest points results in an unorganized cloud of 3D points where the geometric structure of the underlying curves and surfaces is not explicit. This is not much of a problem if the focus is on calibrating the viewpoints as in Phototourism [1]. However, when the 3D object geometry is required, such as in modeling for architecture, archaeology, entertainment, object recognition, and robotic manipulation, it would be useful to augment the output with an explicit geometric structure such as a 3D curve sketch.

The second category of 3D reconstruction techniques, **multi-view stereo methods**, produce detailed 3D reconstructions of objects imaged under controlled conditions by a large number of precisely calibrated cameras [56, 61, 64, 74] (see [134] for a review). However, they cannot handle general scenes, since most approaches are specific to a single object or to a specific type of object, such as buildings. In addition, they often require accurate camera calibration, operate under controlled acquisition, and are often initialized by the visual hull of the object or a bounded 3D voxel volume.

The goal of this work is to augment current multiview reconstruction and calibration technology by developing a generally applicable framework based on **image curves** when a large number of views are available, *e.g.*, a video sequence.

Two fundamental obstacles must be surmounted before image curve fragments can be used in 3D reconstruction and camera calibration. First, given a pair of image curve fragments, each from a different view, there is an ambiguity in intra-curve correspondence, *i.e.*, given a point on one curve, it is not clear which point on the other curve it corresponds to, Figure 5.2(a). This is in contrast to matching two isolated feature points which have no inherent within-pair ambiguity. Second, the linking of edges into curve fragments is not stable across views, even for slight baseline differences, Figure 5.2(b), resulting in multiple transitions as the view changes [79].

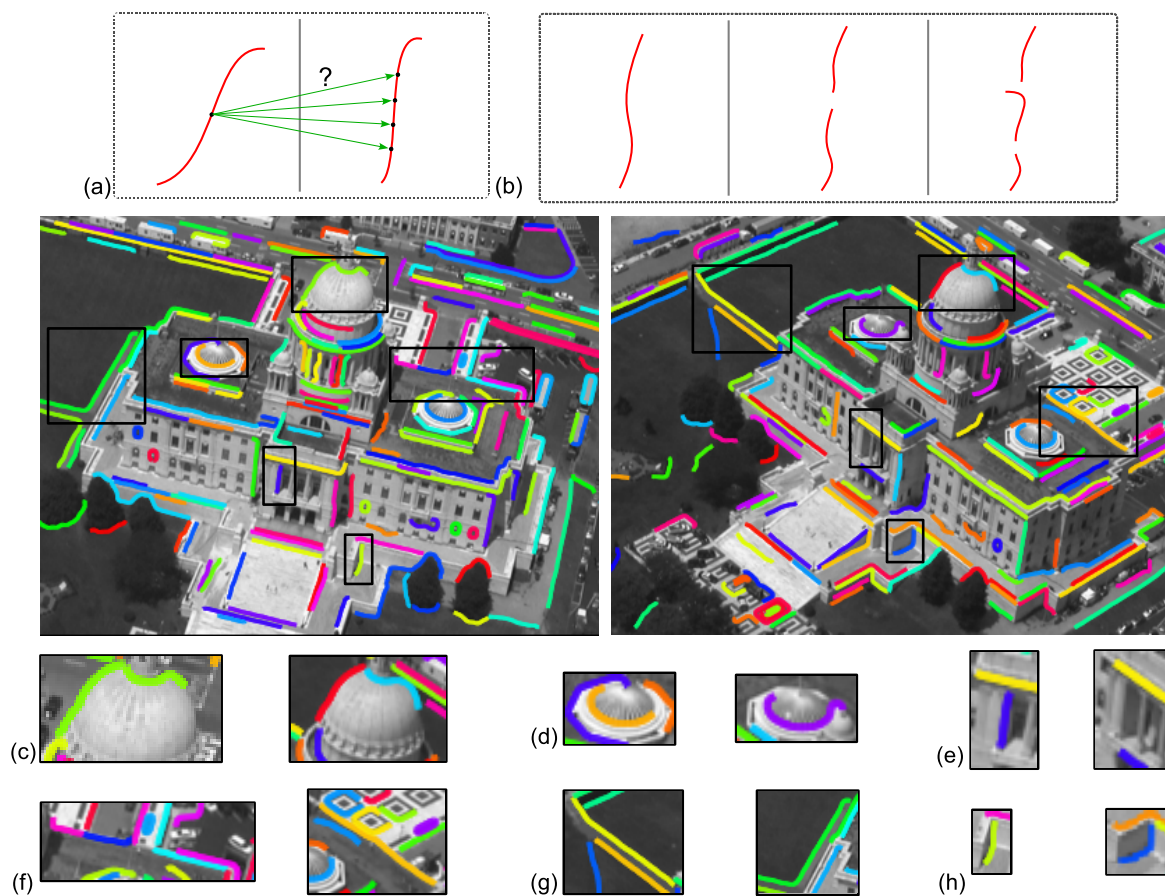


Figure 5.2: Challenges in using curve fragments in multiview geometry: (a) point correspondence ambiguity along the curve, (b) instabilities with slight changes in viewpoint, (c-h) real examples of edge grouping instabilities, such as a curve in one being broken into two in another view, a curve being linked onto background, a curve being detected in one view but absent in another, a curve being fragmented into various pieces at junctions in one view but fully linked in another view, different parts of a curve being occluded in different views, and a curve undergoing shape deformation from one view to the other.

Previous curve-based methods have circumvented problems in various ways. First, the use of general but *closed curves* resolves the curve transition issue [13, 126], but requires the successful segmentation of images which is unlikely. Second, the use of *algebraic curves* resolves the transition issue with low-order models based on extrinsic polynomial equations such as conics [33, 85], quadrics [130, 162], and higher order algebraic curves [86]. However, algebraic curve geometry is restrictive, limiting their application to general scenery. Third, *silhouettes and visual hulls* have been successfully used in several approaches [14, 32, 57, 73, 74, 98], but require a *highly controlled acquisition* for their extraction, and work only for one object per scene. Fourth, *straight lines* are a simple, compact and distinctive geometric representation to be used in generating model hypotheses and detecting objects for man-made environments [38]. However, this approach severely restricts the application domain by not allowing for curved structures. Finally, current multiview methods based on general curves require *highly accurate calibration*, which is not easily available in general.

Overview of our approach: Figure 5.3 illustrates our approach. We assume a large number of views are available, in practice $N \geq 6$ (although 3 views would be the theoretical minimum), *e.g.*, from video or from multiple cameras monitoring a scene. We also assume that the cameras have been *coarsely* calibrated, typically with 2 – 5 pixel error. The goal is to use the collection of image curves in these views to produce a dense collection of unorganized 3D curve fragments, or the **3D curve sketch**, which reflect the underlying geometry arising from a combination of 3D surface geometry and viewing/illumination arrangements, *i.e.*, occluding contours, reflectance contours (albedo discontinuity), shadow curves, shade curves, ridges, *etc.* We also use this reconstruction to refine the cameras so that a better and denser 3D curve sketch can be obtained, and so that a better distinction can be made between occluding contours and view-stationary contours such as reflectance curves and ridges.

Our approach is divided into two stages. In the first stage, where cameras are calibrated coarsely, the goal is to reconstruct a partial, but reliable **core 3D curve sketch** to be used in the second stage for refining calibration and for obtaining a more complete 3D curve sketch. Specifically, pairs of long curve fragments, each from a different view, with sufficient epipolar overlap form a 3D curve fragment hypothesis. For each such pair, the two views from which a curve pair hypothesis is formed are referred to as the **hypothesis views**. The hypothesis views refer to the views relative to that particular hypothesis or set of hypotheses. Any two views from the set of available views usually participate as hypothesis views for some curve pair hypothesis. Each curve pair hypothesis is then reprojected onto a set of other views, the **confirmation views**, and rated for consistency with the image and curve differential geometry. Those hypotheses with sufficient evidence in confirmation views are then reconstructed to form the initial core 3D curve sketch, as discussed in Section 5.2.

The core 3D curve sketch enables a curve-based measurement of calibration accuracy, *i.e.*, by summing up the distances between reprojected curves and supporting image curves. This allows for a refinement of cameras through bundle adjustment, resulting in subpixel calibration errors. The refined cameras in turn allow for additional sensitivity so that smaller curve fragment pair hypotheses can be confirmed or discarded. This is discussed in Section 5.3. Section 5.4 discusses the

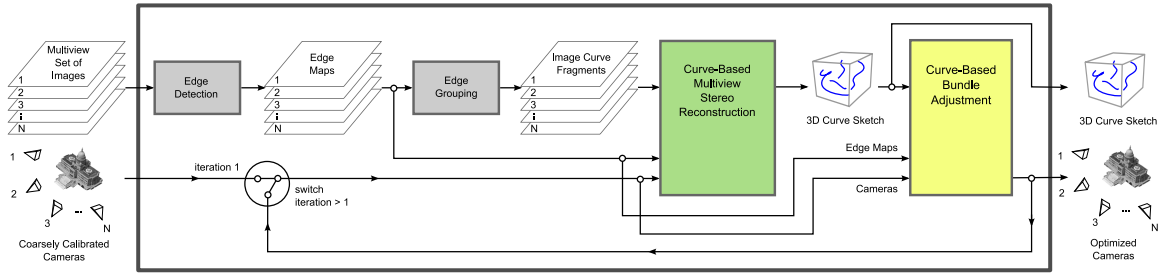


Figure 5.3: A schematic overview of our approach: The input to our system is (i) a set of N images taken of the scene where N is at least 6 but typically larger, and (ii) a set of N coarsely calibrated camera models, one for each view: the system pairs a curve fragment from an “anchor view” with one from another view, “hypothesis views”, to form a curve pair correspondence hypothesis which is verified or discarded based on the edge map of other views, “confirmation views”. Once a set of reliable curve fragment hypotheses are formed, curve-based bundle adjustment is used to refine the cameras and the process is repeated. The output of our system is a set of 3D curve fragments (the 3D curve sketch) and refined cameras.

details of our implementation and Section 5.5 evaluates the approach on several datasets. Figure 5.1 highlights the results.

A critical distinction in reconstructing image curves is whether they arise from *view-stationary* or *view-nonstationary* sources. A *view-stationary* image curve is one arising from the same space curve regardless of the view.

In practice, except for *low-curvature surfaces*, the set of occluding contours cluster in 3D, approximating a single 3D curve. In the first stage of our reconstruction where the calibration errors are high, view-stationary and some view-nonstationary curves are considered together, but they are differentiated in the second stage up to the accuracy of the refined calibration.

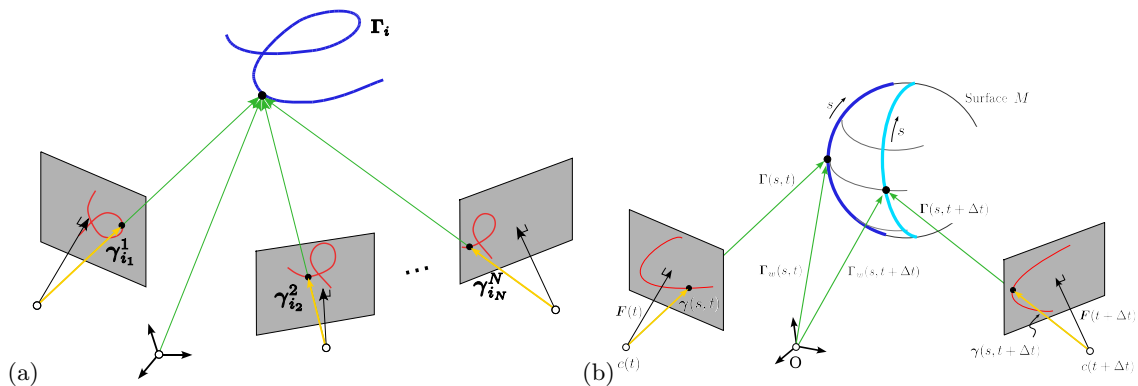


Figure 5.4: A putatively corresponding set of image curve fragments can arise from a single 3D curve.

Additional Bibliography: The present work on the 3D curve sketch lies at the intersection

of different areas of Computer Vision and proposes a new approach, while discussing the merits and drawbacks of numerous existing alternatives. As such, there is a number of previous work to be mentioned. Somewhat related work include Gang Li and Steve Zucker’s work on curves and differential geometry [2, 94, 164], together with Brint and Brady’s binocular curve matching system using heuristic shape similarity constraints [17]. These systems, however, don’t use a large number of views nor work under the flexible settings of uncontrolled acquisition with coarsely calibrated views. Older papers that can be related to our approach are [3, 7, 8]. Our approach can be thought of as an automated version of the curve-based CAD system from [159].

The insight behind the robustness of our support function (described in Section 5.2), that of integrating the number of inliers, can be remotely traced back to the generic robust model fitting approach of RANSAC [54] that has been the core of many successful approaches to computer vision, such as automatic structure from motion. The idea, coupled with differential-geometric constraints, allows our matching method to be extremely robust to missing edgels and to outliers.

It may also be worthwhile mentioning that our approach is still within the tradition of using features for structure and motion [145], and bears most of their advantages and motivation; the novelty lies in bringing the richer curve fragments to practice, instead of corners or SIFT features.

Additional citations include [10, 23].

5.2 Curve-based Multiview Stereo

The approach presented here is based on reasoning with image curve fragments obtained from a large number of views. We denote image curve fragments by γ_i^v , where v indexes into views, $v = 1, \dots, N$, and i enumerates the image curves within each view v , $i = 1, \dots, M_v$. The goal is to produce a set of 3D stationary curve fragments $\{\Gamma_1, \Gamma_2, \dots, \Gamma_K\}$, supported by sufficient evidence from the multitude of images. Several observations are important. First, a given 3D curve fragment Γ_k may not project to all views due to occlusion or due to limited field of view. Thus, evidence for Γ_k cannot be expected in all views. Second, in views where Γ_k is visible it may only be partially visible, or if fully visible, there may not be sufficient image evidence to produce image curves. Thus, the projected image curve corresponding to Γ_k may be partially visible or it may be broken into segments. Furthermore, the partitioning of the ideally projected curve into segments can differ substantially from view to view, Figure 5.2(b). Third, even if Γ_k is fully visible as a whole image curve fragment in each view, the intra-curve correspondence between two projected image curves is not clear, Figure 5.2(a). While the above problems are generally expected from *some* image curve projections of a space curve, there is typically at least a few views where there is partial correspondence.

Assumption 5.2.1. For each stationary 3D curve fragment there exist at least two views v_1 and v_2 for which a substantial subsegment of the 3D curve fragment Γ_k projects onto portions of extracted image curve fragments $\gamma_{i_1}^{v_1}, \gamma_{i_2}^{v_2}$, Figure 5.5(a).

Since a point in one view can correspond to a point in another view only if they each lie on corresponding epipolar lines, and since an image curve segment is an ordered collection of points,

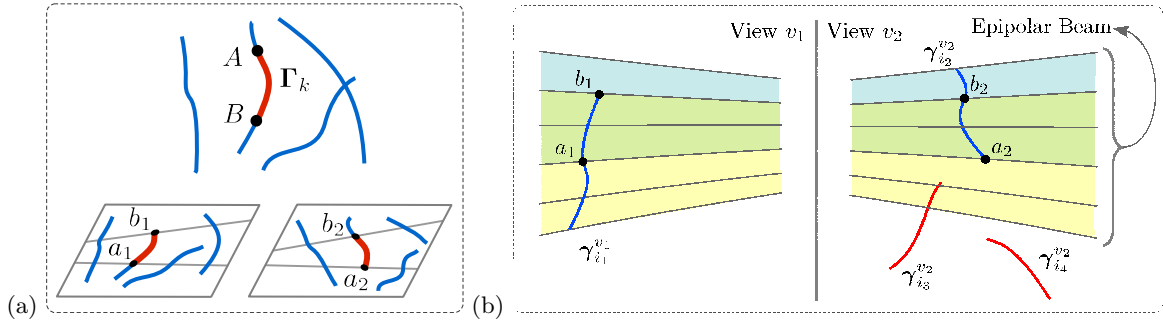


Figure 5.5: (a) Each space curve, or at least a significant subsegment of it, is expected to have corresponding image curve subsegments in at least two views. (b) An “epipolar band” for a curve segment is the set of epipolar lines going through points of that curve segment. The epipolar band of image curve $\gamma_{i_1}^{v_1}$ (green and yellow) delineates potential image curve matches by requiring some overlap in this region, allowing for $\gamma_{i_1}^{v_2}$ and $\gamma_{i_3}^{v_2}$ but not $\gamma_{i_4}^{v_2}$. Conversely, the epipolar band of $\gamma_{i_2}^{v_2}$ (green and blue) selects $\gamma_{i_1}^{v_1}$, and others. The correspondence between $\gamma_{i_1}^{v_1}$ and $\gamma_{i_2}^{v_2}$ is restricted to the subsegments a_1b_1 and a_2b_2 , respectively, *i.e.*, the common (green) epipolar band, required to exceed a minimal epipolar overlap ($\tau_e = 5$ pixels).

a curve segment in one view corresponds to a curve segment in another view if they lie on a corresponding set of epipolar lines, which can be referred to as an epipolar band, *e.g.*, the green region in Figure 5.5(b). The common portions of such pairs of curves are potential candidates for having arisen from a single common 3D curve fragment.

Definition 1. Two image curve fragments $\gamma_{i_1}^{v_1}$ and $\gamma_{i_2}^{v_2}$ with portions lying on a common epipolar band, $\tilde{\gamma}_{i_1}^{v_1}$, $\tilde{\gamma}_{i_2}^{v_2}$, respectively, form a **curve fragment hypothesis** whose reconstruction Γ_k represents a potentially valid 3D curve hypothesis.

It is clear that not all curve fragment pair hypotheses are valid. In fact, the ratio of valid hypotheses to spurious ones is roughly on the order of the number of image curve fragments sharing an epipolar band with each given curve, typically ranging from ten in sparse scenes to dozens or hundreds in more complex scenes. This ambiguity can be resolved by confirmation of hypotheses in other views, comparing the projection of Γ_k onto another view v , namely $\tilde{\gamma}^{k,v}$, by consulting image evidence. While it may seem natural to demand the presence of an image curve γ_i^v that closely resembles $\tilde{\gamma}^{k,v}$, this is not a realistic expectation due to instabilities in the edge linking process, Figure 5.6.

Alternatively, the edge map itself can be interrogated, to avoid exposure to the frailty of the edge linking process, as is done in object recognition strategies [119, 139, 149] in curve tracking [79], and in satellite sensor registration [124]. However, an edge without proper local context supports numerous potential curve hypotheses. As such, in the presence of clutter and texture where numerous edges arise, multiple incorrect hypotheses receive sufficient support. Thus, an edge without additional

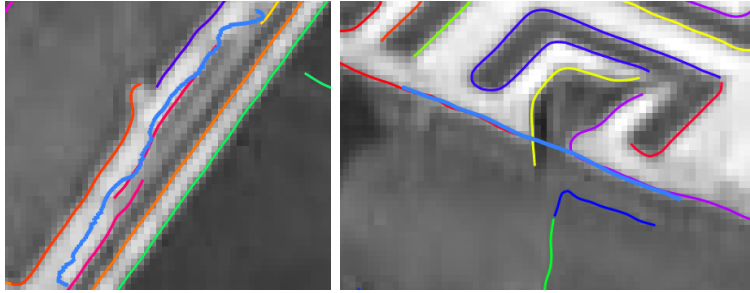


Figure 5.6: Real examples of how instabilities in the edge linking process make it difficult to evaluate the support of a reprojected curve (light blue) using curve fragments (random colors).

local support cannot reduce ambiguity and is not significantly selective.

The natural compromise between long image curves, which are inherently unstable and unreliable, and short edge elements, which are easy to compute but are inherently not sufficiently selective of curve hypotheses, is to use *short curve segments*. The advantage is that short curve segments are relatively more stable with variations and have sufficient local context to be selective. Their drawback is that many expected short curve segments may not be produced due to incorrect grouping.

The above dilemma can be resolved by considering all possible reasonable local groupings of edges into small curve segments. The use of multiple curve segments hypotheses passing through each edge introduces clutter, but the use of other views ensures that erroneous groupings are not validated towards this end. This avoids missing out on groupings due to poor edge linkers while dealing effectively with clutter. Specifically, Tamrakar and Kimia [144] developed a geometric validation scheme to retrieve all feasible local groupings of edges in a small neighborhood, say 7×7 . They show that a single edge hypothesizes a vast space of curves, while a pair of edges hypothesize a reduced space, a triplet of edges represent a more restricted set of curves, *etc.*, Figure 5.7. Such a combinatorial analysis is feasible and results in a limited set of “discrete curvelets” for each edge. It is this discrete curvelet that is the natural compromise between unstable long curves and context-less local edges, Figure 5.8. Figure 5.9 shows the reprojection of a 3D curve hypothesis onto the set of possible curvelets at one point.

The image support for a given reprojected curve $\tilde{\gamma}^{k,v}$ is therefore based on summing the support of edges which sufficiently support this hypothesis. An edge supports a projected curve only if there exists a discrete curvelet sufficiently supporting it. This means that each edge must be sufficiently close and be of the right orientation. Specifically,

Definition 2. (*Weakly supporting edge*) An edge at point p_0 with tangent t_0 of an edge map M

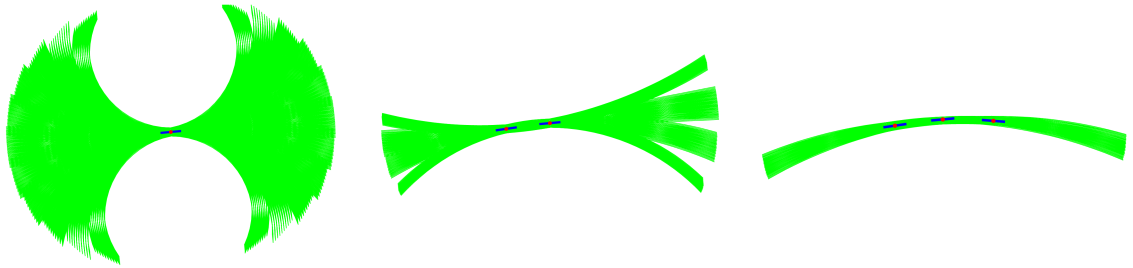


Figure 5.7: A single edge hypothesizes a vast space of curves, while a pair of edges hypothesize a reduced space, a triplet of edges represent a more restricted set of curves, *etc.*

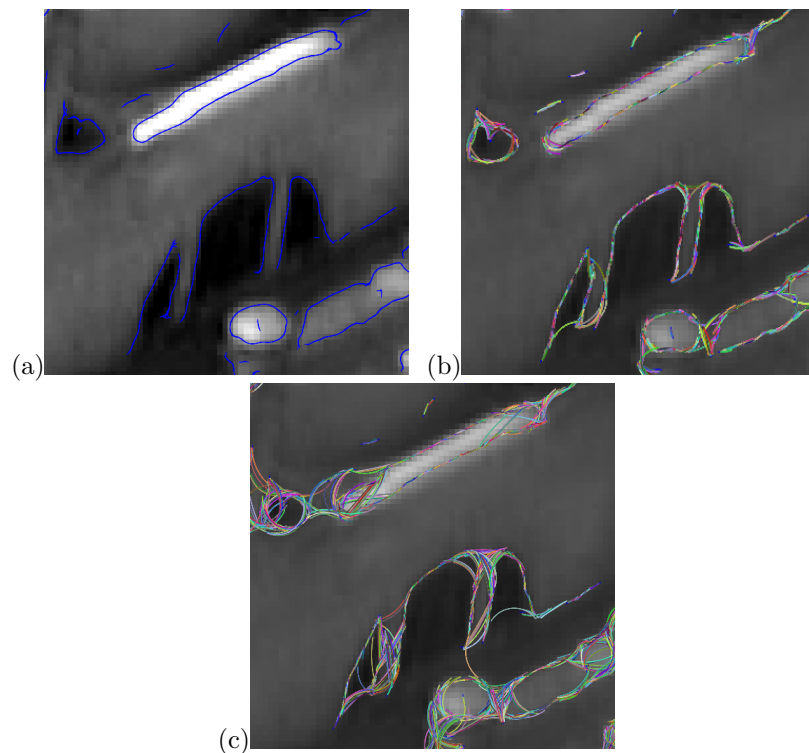


Figure 5.8: An example of curvelets detected in a real image (a side window of the Capitol building depicted in Figure 5.2): (a) The original edges; (b) all the curvelets representing local groupings in a 5×5 neighborhood, and (c) all the curvelets representing local groupings in a 7×7 neighborhood.

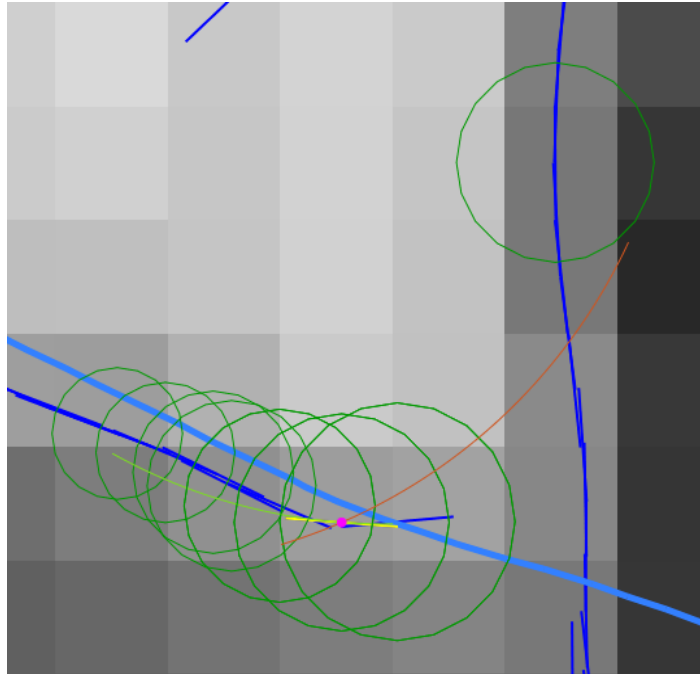


Figure 5.9: Among the discrete curvelets (red and light green) at an edge point (yellow with magenta dot), only one (green) matches the reprojected curve $\gamma^{u,v}$ (light blue). The remaining edges in the edge map are shown in darker blue.

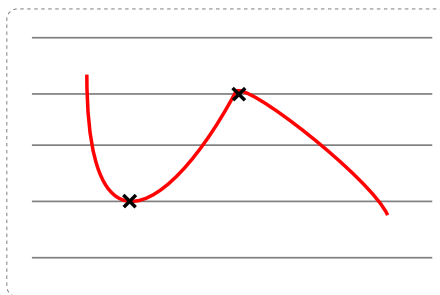


Figure 5.10: We avoid multiple intersections with epipolar lines by breaking curve fragments at epipolar tangencies.

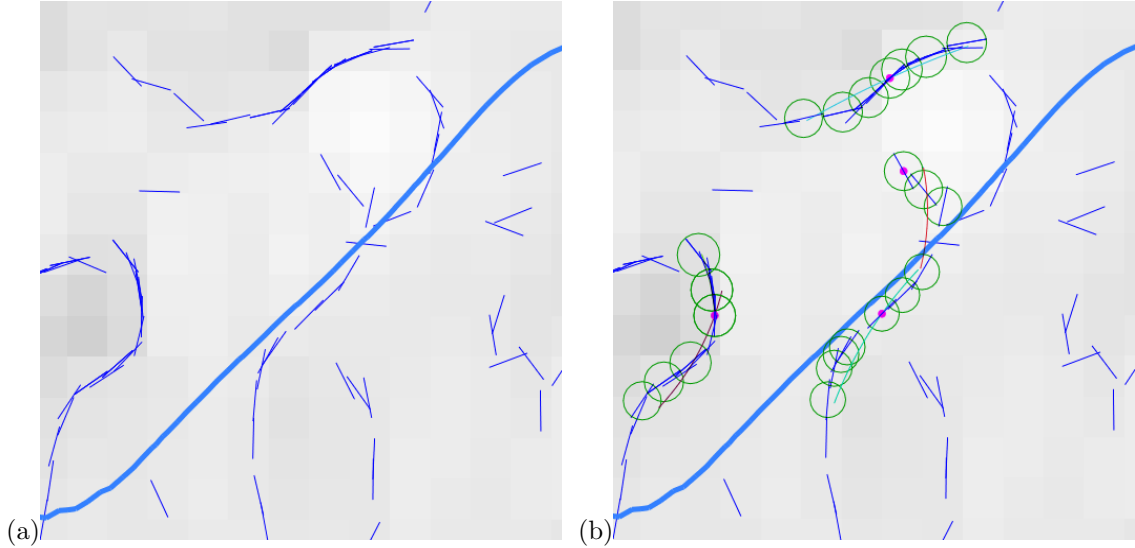


Figure 5.11: The instability of the edge linking process can be altogether avoided if we use edges, as done in recent object recognition strategies, *e.g.*, oriented chamfer distance [119, 139, 149]. However, reprojected contours (blue) can at times get significant support from cluttered or textured areas (a). This motivates the exclusive use of edges for which there exists a local grouping consistent with the reprojected curve (b). Two of the edges with a magenta dot in (b) support the reprojected curve with a curve model (green) while two others do not (red).

supports an an image curve point $\gamma(s)$ with tangent $\mathbf{t}(s)$ when:

$$\begin{cases} \mathbf{p}_0 = \operatorname{argmin}_{\mathbf{p} \in M} d(\mathbf{p}, \gamma(s)) & (5.2.1) \\ d(\mathbf{p}_0, \gamma(s)) < \tau_d & (5.2.2) \\ \angle(\mathbf{t}_0, \mathbf{t}(s)) < \tau_\theta, & (5.2.3) \end{cases}$$

where $d(\mathbf{p}, \gamma(s))$ is the Euclidean distance between a point \mathbf{p} and the curve point $\gamma(s)$.

This is a minimum requirement, however. It is additionally expected that the curve is supported for at least one discrete curvelet going through that edge. Specifically, an edge is supportive of a reprojected curve if it is close to it and of the right orientation, and there is a local grouping where the majority of the edges are weakly supporting.

Definition 3. (*Strongly supporting edge*) An edge strongly supports a curve if (a) it is a weakly supportive edge and (b) it participates in a discrete curvelet where τ_s percent (typically 70%) of the edges are weakly supporting edges.

The use of tangent orientation is significant since a reprojected curve on clutter would otherwise receive high support. This is confirmed by our experiments since performance drops when τ_θ is increased. The total support for a reprojected curve is the sum of support from all supporting edges.

Let the number of strongly supporting edges of a curve point $\gamma(s)$ in view v be denoted $\phi(\gamma(s))$. Then, the total support S a hypothesis w_k has from view v is

$$S_v(w_k) = \int_{s=0}^{L_k^v} \phi(\gamma^{k,v}(s)) ds, \quad (5.2.4)$$

where L_k^v is the length of $\gamma^{k,v}$. A view is considered a *supporting view* when support for a view v that a curve is above τ_v .

Considering the potential for occlusion, failure of edge linking, *etc.*, evidence is solicited from more than one confirmation view for each curve fragment pair hypotheses by summing support over supporting views

$$\mathcal{S}(w_k) = \sum_{v \neq v_1, v_2}^N [S_v(w_k) > \tau_v] S_v(w_k). \quad (5.2.5)$$

A space curve hypothesis Γ_k is considered validated if the extent of this support exceeds threshold τ_t .

A two-stage process. The process first considers reliable curves, which reduces ambiguity, and then in a second stage, after a calibration refinement, considers a larger set of curves to produce a denser 3D curve sketch.

In the first stage, only sufficiently long curve fragments (typically of length $l > \tau_l = 40$ pixels for HD video) are considered, since they are reliable indicators of image structure. The goal in the initial stage is to produce a **core** set of reliable curves Γ_k which can be used to bootstrap a better calibration. Since long curves typically undergo a wide range of transitions with view changes, only a minimal but sufficient epipolar overlap ($\tau_e = 5$ pixels) is required. Once better calibration is available, the length condition is relaxed for the subsequent stages thus producing a richer 3D curve sketch. Multiple curve intersections are handled by breaking all curve fragments at epipolar tangencies, Figure 5.10, and keeping the ones with length greater than τ_l . The result is a set of curve fragment pair hypotheses $\mathcal{W} = \{w_k, k = 1, \dots, K\}$ collected from pairs of hypothesis views, Figure 5.14.¹

The set of curve fragment pair hypotheses \mathcal{W} is now thresholded for extent of support τ_t . However, observe that an image curve fragment γ_1^1 may pair with more than one curve fragment in view 2, due to epipolar ambiguity, *e.g.*, (γ_1^1, γ_1^2) and (γ_1^1, γ_2^2) may be both possible hypotheses, exceeding a threshold of support, but both cannot simultaneously be valid. Thus, we use a greedy assignment to resolve conflicts: the curve fragment pair hypotheses are rank-ordered and iteratively the highest-ranking hypothesis removes all conflicting lower-ranking hypotheses.

A ratio parameter τ_r is used in our system to prune out unreliable matches. The insight is that we discard a curve fragment match hypothesis if the *second best* match to a given curve γ^1 has support s_2 close to the support s_1 of the best match, that is:

$$\tau_r \cdot s_2 > s_1 \quad (\text{condition for ambiguous match}). \quad (5.2.6)$$

¹Robert and Faugeras [131] (and later Schmid and Zisserman [133] using the Trifocal tensor) also use differential geometry in a trinocular setting, but these systems require extremely well-calibrated cameras, are restricted to 3 views, and employ additional heuristic constraints.

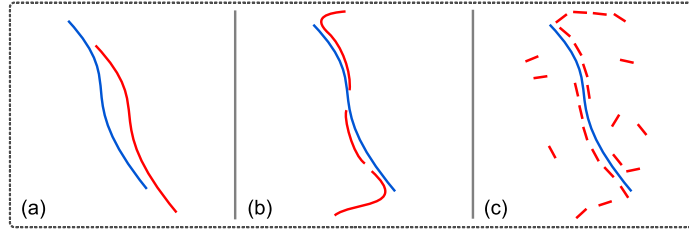


Figure 5.12: The reprojection of a curve fragment pair $\gamma_{i_1}^{v_1}, \gamma_{i_2}^{v_2}$ onto a third view (shown in blue) can enjoy the support of an existing curve fragment γ_i^v (red) as in (a). However, much more typically, due to the instability of the edge linking process, it is more likely to expect support from several pieces of curve fragments as in (b), with missing segments. This instability motivates the use of a small neighborhood of an edge – not entire curve fragments – in the confirmation process.

This is similar to the criterion used to discard unreliable SIFT correspondences in point-based structure from motion. We use it to overcome the repeated structures problem without having to use a large number of confirmation views N_c .

It should be emphasized that while this process may lead to numerous false negatives in forming curve fragment pair hypotheses (especially in the first stage when the threshold of support is higher for reliability *i.e.*, a curve fragment in one view may break up differently in a second view), this is not a major concern because the set of corresponding curve fragments $\{\gamma_{i_1}^1, \gamma_{i_2}^2, \dots, \gamma_{i_n}^n\}$ need only be represented by one curve fragment pair. Thus, if a veridical curve fragment pair is not selected in hypothesis views 1 and 2, it is likely to be selected in another pair of hypotheses. In fact, the confirmed curve fragment pair hypotheses induces a grouping on the remaining views. This notion of interacting 3D curve reconstruction and edge linking is actually a significant departure from purely bottom-up edge linking approaches, but its development is beyond the scope of this thesis.

View Selection. While ideally the set of curve fragment pair hypotheses \mathcal{W} should be formed from all pairs of hypothesis views and tested on all the remaining ones, this is not practical when the number of views is in tens or hundreds, as in a video sequence, thus motivating a **view selection strategy**. On the one hand, the reliability of reprojection onto a third view is directly related to how close the two views are: reprojections from two views with a small baseline are not stable. On the other hand, pairs of views with a large baseline may not have many image curves with a common source due to occlusion. Our experiments with real data indicate that a baseline of 40° leads to stability comparable to calibration accuracy. Beyond this value of baseline, the number of hypothesis views is only constrained by the extent of computations that can be afforded. We select pairs of consecutive views $b = 40^\circ$ apart as hypothesis views. An identical strategy holds for selecting confirmation views. The only difference is that in the confirmation process visibility is more of an issue than reconstruction stability. Thus, the selection follows a smaller baseline $b_c = \frac{1}{3}b$, in the vicinity of the hypothesis views but excluding the selected hypothesis views.

5.3 Camera Calibration Refinement

The goal of this section is that given the core 3D curve sketch $\{\Gamma^1, \dots, \Gamma^k\}$ obtained after the first stage, to refine the calibration by minimizing reprojection error.

Let the camera parameters for each view be $P_i = \mathcal{K}_i [\mathcal{R}_i | \mathcal{T}_i]$, $i = 1, \dots, N$, where \mathcal{K}_i , \mathcal{R}_i and \mathcal{T}_i are the intrinsic parameter matrix, the rotation matrix and the translation vector, respectively. Let the set of validated curve fragment pair hypotheses $\overline{\mathcal{W}}$ underlying the core 3D curve sketch be

$$\overline{\mathcal{W}} = \{w_k = (\gamma_{i_1}^{v_1}, \gamma_{i_2}^{v_2}), k = 1, \dots, K\}. \quad (5.3.1)$$

These hypotheses $\overline{\mathcal{W}}$ together with camera parameters for all views $\mathcal{P} = (P_1, P_2, \dots, P_N)$ reconstruct the 3D curve sketch $\{\Gamma_1, \Gamma_2, \dots, \Gamma_K\}$. Assuming the set of curve pair correspondences $\overline{\mathcal{W}}$ is correct, the curve-based reprojection error, *i.e.*, the degree the reprojected Γ_k , $\gamma^{k,v}$ agrees with its edge evidence in that view M_v defines the quality of the calibration. Formally, define the reprojection error of a 3D curve fragment Γ_k arising from views v_1 and v_2 as

$$f_{k,\mathcal{M}}(\mathcal{P}) = \sum_{\substack{v=1 \\ v \notin \{v_1, v_2\}}}^N \int_0^{L_u} d(\gamma^{k,v}, M_v) ds, \quad (5.3.2)$$

where $d(\gamma_0^{k,v}(s), M_v)$ is the distance between the point $\gamma_0^{k,v}(s)$ and the line containing the closest oriented edge point in M_v . Note that the closest edge point must satisfy the distance and orientation thresholds, τ_d and τ_θ , as in equations (2) and (3) respectively; oriented distance takes into account both distance and orientation and is defined as total reprojection error from the reliable, core 3D curve sketch as

$$f_{\overline{\mathcal{W}},\mathcal{M}}(\mathcal{P}) = \sum_{k=1}^K f_{k,\mathcal{M}}(\mathcal{P}). \quad (5.3.3)$$

We then seek cameras that minimize the error

$$\hat{\mathcal{P}} = \underset{\mathcal{P}}{\operatorname{argmin}} f_{\overline{\mathcal{W}}}(\mathcal{P}), \quad (5.3.4)$$

The independence of curve projections from each other and the limited visibility of a curve across views implies a sparsity that allows for a successful implementation of the optimization using Levenberg-Marquardt as used in the traditional bundle adjustment. The total calibration error is reduced to subpixel accuracy, as illustrated in Figure 5.13.

5.4 Implementation

Edge and curve fragment detection: The integration of curve content over many views when clutter is present relies heavily on the orientation of curve tangents in addition to curve position. Thus, the edge detector used must yield very reliable orientation estimates. We use the subpixel edge map produced by a third-order operator [144], which explicitly addresses the orientation reliability,

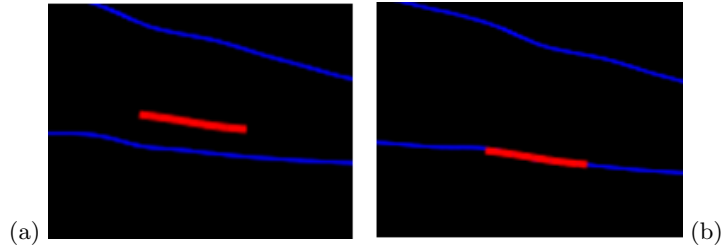


Figure 5.13: Example of a projected curve $\gamma^{k,v}$ (red) on edge map M_v (blue) before calibration refinement (a) and after (b).

and their symbolic edge linker which addresses the grouping of edges into curve fragments. Examples are shown in Figure 5.14.

Initial calibration. Initial calibration is obtained using traditional automatic structure from motion techniques [1, 125], or if that fails, by manually specifying corresponding points and solving for the cameras. The initial calibration is typically imprecise (1-6 pixels reprojection error), which we call “coarse” calibration. We assume intrinsic parameters to be precisely known, with the imprecision residing on the extrinsic parameters. We are also in the process of developing a curve-based autocalibration system to augment point-based autocalibration using the ideas presented here.

Epipolar line intersection. The breakup of curve fragments at points of epipolar tangency implies that each segment can be parametrized by epipolar angle, *i.e.*, the angle of the epipolar line at each curve sample with the x axis. This allows for a rapid test of whether an epipolar line intersects a curve by simply looking up the two curve boundary epipolar angles. Such an operation is necessary for determining epipolar candidates/epipolar overlap, and for 3D reconstruction and reprojection by establishing pointwise correspondence within the curves. Furthermore, the intersection of an epipolar line and a given curve is reduced to simply checking which discrete parameter interval of the curve the epipolar line belongs to. The removal of points near epipolar tangency by insisting that all curve tangents have an angle of τ_α or more ($\tau_\alpha = 10^\circ$) also allows for a stable intersection point computation.

Curve reprojection. The 3D curve fragment Γ_k is reprojected onto view v . We can also use an image to image transfer (*e.g.* the Trifocal transfer [69]), but we followed the simpler implementation.

The distance $d(\gamma^{k,v}, M_v)$ uses the distance transform of edge maps to lookup the closest edgels to each point of the curve. First, the image edgels are hashed into a bucket grid so that subpixel edgels can be very efficiently indexed. Typically, the bucket size is that of a pixel. The distance transform gives the distance of any bucket to the nearest bucket containing an edgel, and associated nearest label map provides the address of this nearest bucket. Thus, for each projected curve point one can quickly lookup the nearest bucket containing an edgel. However, instead of using the distance transform values, we directly measure the distance between the query point and the subpixel edgel inside the nearest bucket. The distance transform and nearest label map are efficiently precomputed for the entire set of images prior to matching and calibration.

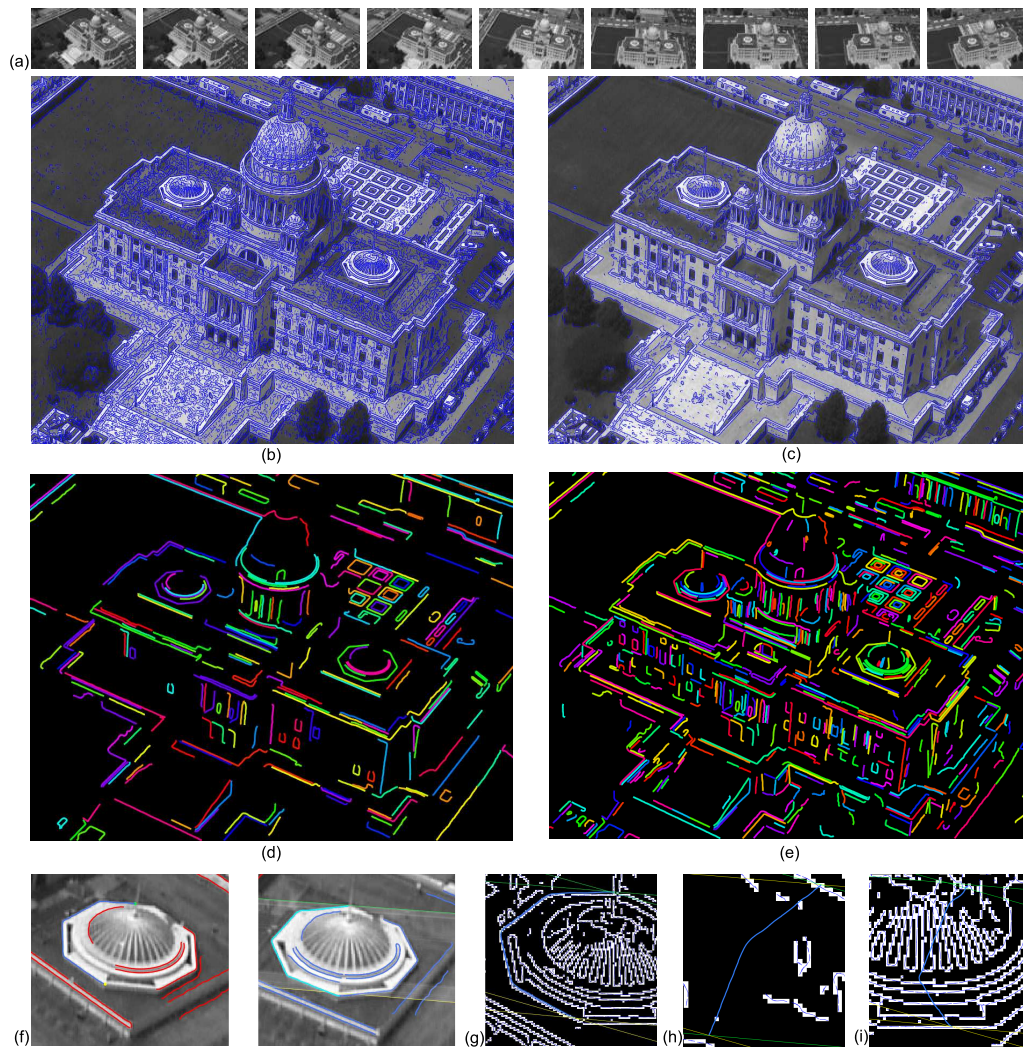


Figure 5.14: (a) A set of sample views from the “Capitol Sequence”; (b) detected subpixel edges at low gradient threshold are used to generate hypothesis curves; (c) detected subpixel edges at a higher gradient threshold are used in the confirmation views; (d) curve fragments with length > 40 pixels used in Stage I; (e) fragments with length > 20 pixels; (f) a selected curve is highlighted in view 1 and the curves in view 2 with sufficient epipolar overlap are marked in blue. In the next three images the reprojected hypothesis curve pair is shown in cyan, confirmed in (g) but rejected in (h) and (i). The example in (i) shows the need for orientation in the presence of clutter!

Parameters of the system. Table 5.1 gives a description of each parameter of the system.

Symbol	Description	Default Value
τ_l	minimum length of image curve fragments	40 pixels (first stage), 20 pixels (second stage)
τ_e	minimum epipolar overlap	5 pixels
τ_d	maximum distance of supporting edgels	10 pixels
τ_θ	maximum orientation difference of supporting edgels	10°
τ_α	minimum angle of curve segments with epipolar lines to reconstruct	10°
τ_t	minimum total inliers for a match to be considered reliable	5 edgels
τ_v	minimum number of inliers of a supporting view	10 edgels
τ_r	minimum best to second best ratio	1.5
b	baseline between hypothesis views	40°
b_c	baseline between consecutive confirmation views	$\frac{1}{3}b$
N_c	number of confirmation views	4

Table 5.1: Table of parameters of the system.

Distance Transform. We used one of the fastest exact Euclidean algorithms for our distance transform, instead of an inexact Chamfer distance map [43, 105]. This is crucial for the correctness of our approach.

Language. We used C++ to implement the system based on widely-available open source libraries, such as Boost (www.boost.org), VXL (vxl.sourceforge.net), and others. The experiments ran on Linux but the code is very portable.

5.5 Experimental Results

The Capitol Building Sequence consists of 500 frames covering a 90° helicopter fly-by from the Rhode Island State Capitol, using a High-Definition (HD) camera (1280×720). Intrinsic parameters were obtained from Matlab Calibration toolbox. Extrinsic parameters were obtained by manually marking/tracking 30 corresponding points on every view they appear in. This data is challenging due to a mix of curved and linear structures, a complex background, and structures of various sizes.

The Capitol Building High Sequence consists of 256 frames coming from a 270° helicopter fly-by at higher altitude capture the Rhode Island State Capitol. Objects of interest appear in lower resolution than in the regular Capitol sequence, leading to higher reconstruction errors.

The Downtown Sequence consists of 173 frames covering a 360° helicopter fly-by over downtown Providence, using an HD camera (1280×720). The sequence was automatically calibrated using Bundler [1], a SIFT-based structure from motion program. This data contains correlated and repeated structures, *e.g.*, a large number of long straight lines, easily confused with other lines due a to lack shape.

The Dinosaur Sequence from [134] was selected because it is a standard multiview-stereo dataset used by the computer vision community, even though it is not representative of the types of general

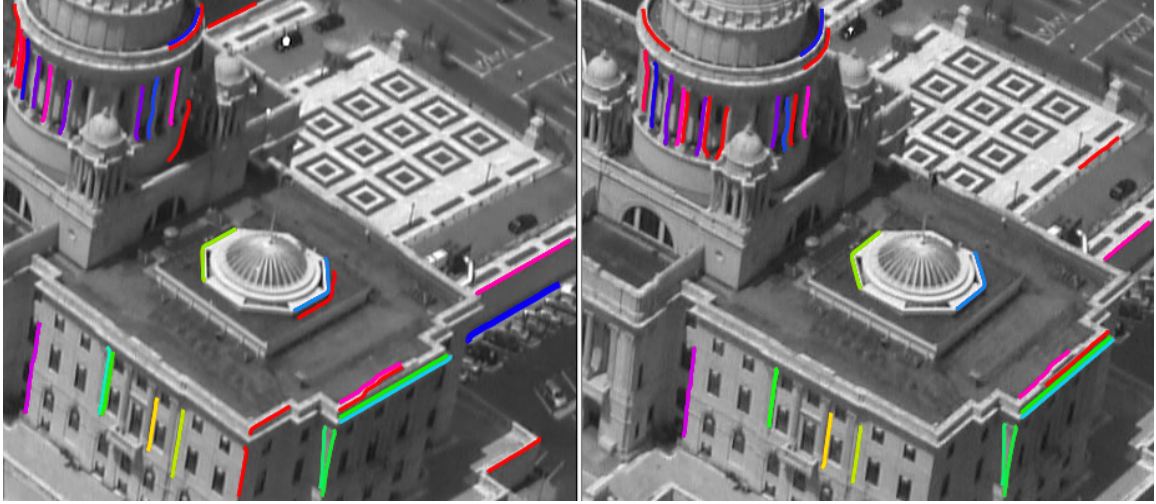


Figure 5.15: Example of human-marked ground truth correspondences on a small section of the “Capitol Sequence”. Curves that were marked as corresponding are represented with the same color, and curves not participating in any correspondence are in red. The thicker blue curve in the left image indicates a curve the user is inspecting.

scenery for which our approach can be most useful. The Cameras sample 363 views at 640×480 on a hemisphere around the object. The data is low resolution and contains a large number of occluding contours.

Ground truth for the 3D Surfaces is only available for the simplest scenes, such as the Dinosaur sequence and others from [134]. It is virtually impossible to get comprehensive ground-truth for the challenging real-world datasets tested in the present work. Even if such a ground truth were available, the evaluation of the system as a whole would confound the performance of the proposed approach with that of the edge detector and linker. Rather, we focus instead on evaluating the ability of the system to find the correct correspondence between curve fragments in two views, by manually recording the ground-truth correspondence between two views of each dataset. Figure 5.15 shows an example of such ground-truth.

A GUI tool was developed for this purpose to insert, delete, and inspect correspondences between two views. The views selected for this were such that they had a baseline near the minimum required for accurate 3D reconstruction for the given dataset. Curve fragments were detected on each hypothesis view and broken at epipolar tangencies. The human operator clicks on each curve fragment in view 1, and the GUI tool highlights the candidate curves in image 2 having epipolar overlap.

The operator is instructed to mark as ground truth correspondents *all* the curve fragments which come from the same underlying physical 3D structure, *i.e.*, having subsets coming from the same 3D positions. If there is any ambiguity in this selection, the operator is allowed to use every information possible to try to disambiguate the match: she is allowed to hypothesize a match in view 2 and the

software will display reprojections for that match in all other confirmation views; she can also overlay the curve fragments onto the original images. The user can then gather evidence until the hypothesis is clearly decided. If all fails, the user can request to reconstruct the hypothesis and examine it in 3D; she can also reconstruct all the unambiguous groundtruth marked so far and examine how well the reconstructed 3D curve hypothesis fits in the context of the already reconstructed ground truth. This kind of reasoning leaves little doubt as to what the corresponding curves should be. Ambiguity can still remain, such as a few very closeby and similarly shaped curves in a low-spatial frequency section of the image. In such cases, the user is allowed to mark two alternatives as equally good matches, but these form a small percentage of the total data. Positive matches are the pairs (γ_i^1, γ_j^2) marked by the human in the ground truth. Conversely, negative matches are all the pairs not marked in the ground truth.

Results. Figure 5.16 shows Precision/Recall curves for each dataset comparing the core 3D curve sketch to the ground truth. All datasets get 100% precision at 1/3 recall, which is 40 curves for the Downtown, 65 curves for the Capitol sequence, and 30 curves for the Dino sequence. This is already plenty enough for calibration (much more than the usual 40 corresponding points needed for stable estimation [69]). Note that these numbers of core reliable curves are much higher if we pick the right operating point *per* dataset.

The role of differential geometry, or orientation, in this work, is explored by enlarging the orientation threshold to 90° so that it no longer plays a role. The plots on the right of Figure 5.16 show a significant degradation as the role of orientation is reduced. We expect a similar role for curvature.

Figure 5.17 shows the quality of the core 3D curve sketch which is obtained by integrating a large number of hypothesis views. This is best viewed in 3D (see the supplementary data of the paper [47]) to appreciate the 3D structure presented by the 3D curve sketch. Observe the potential of this curve sketch as a scaffold on which surface patches can be constructed. Other results are shown in Figures 5.18, 5.19, and 5.20.

Running times. Tables 5.2 lists the total running time of our research code for the system, and Table 5.3 compares the running times of our subpixel edge detector plus linker versus SIFT.

5.6 Conclusion

We have presented a novel framework for multiview reconstruction and calibration refinement based on image curve content. The approach augments existing interest-point based and stereo approaches in providing explicit curve geometry as well in extending applications where the assumption of these methods fail but image curve content is present. A key capability is integration across many views, *e.g.*, as in Google’s Street View. The present work is expected to form the initial building block in a broader effort to use image evidence of the explicit geometry of curves and surfaces and reconstruct these by integrating information across many views. The 3D curve sketch presented here, when enriched by interpolating across epipolar gaps, will then be the initial scaffold on which surfaces

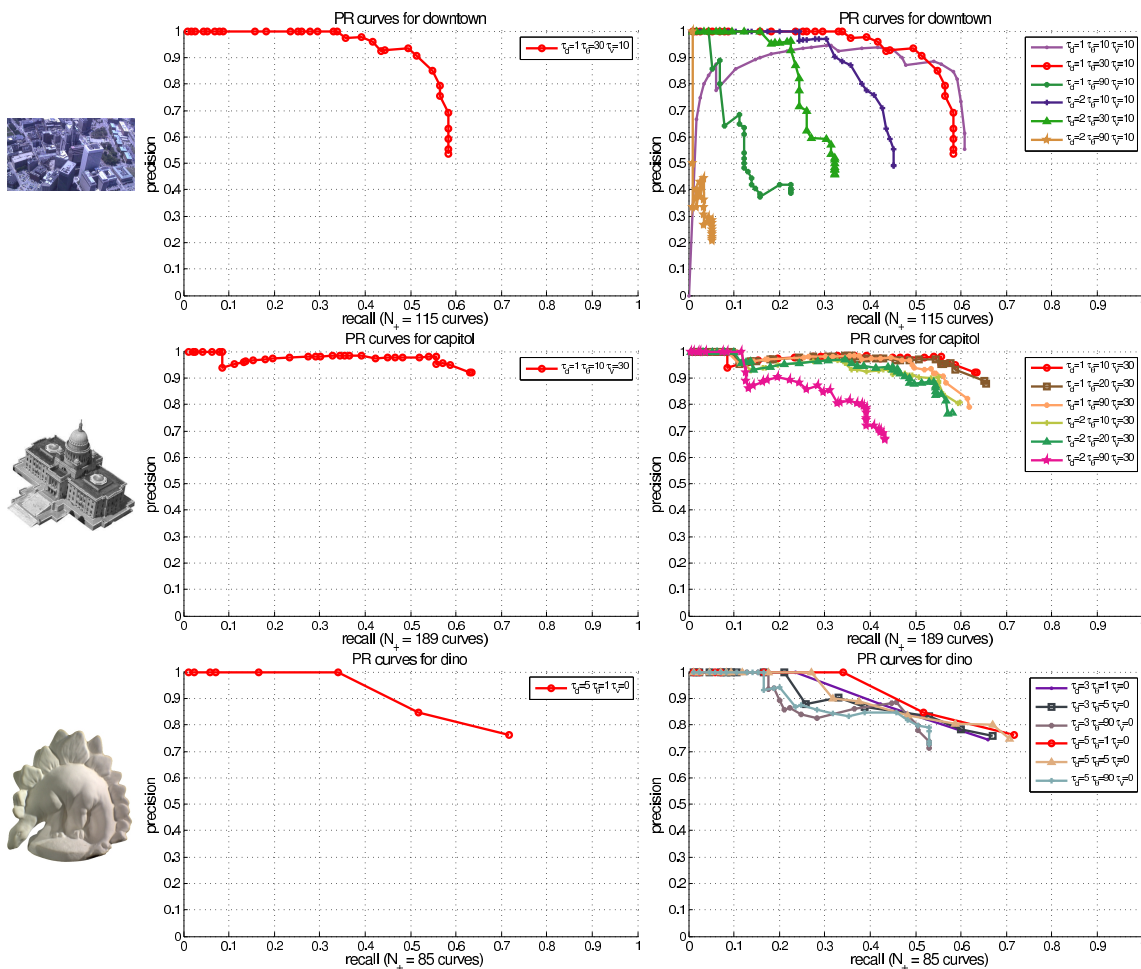


Figure 5.16: Precision/Recall curves for the core 3D curve sketch correspondence, obtained by varying the total support score τ_t . **Left:** the optimal evaluation curve shown in red; **Right:** varying the orientation threshold τ_θ to demonstrate the usefulness of the differential geometry constraint. Notice how eliminating orientation ($\tau_\theta = 90^\circ$) degrades performance. These experiments used disambiguation threshold $\tau_r = 1.5$, length $\tau_l = 40px$ for the Downtown and Capitol sequences (hi-res), and $\tau_l = 20px$ for the Middlebury Dino sequence (low-res).

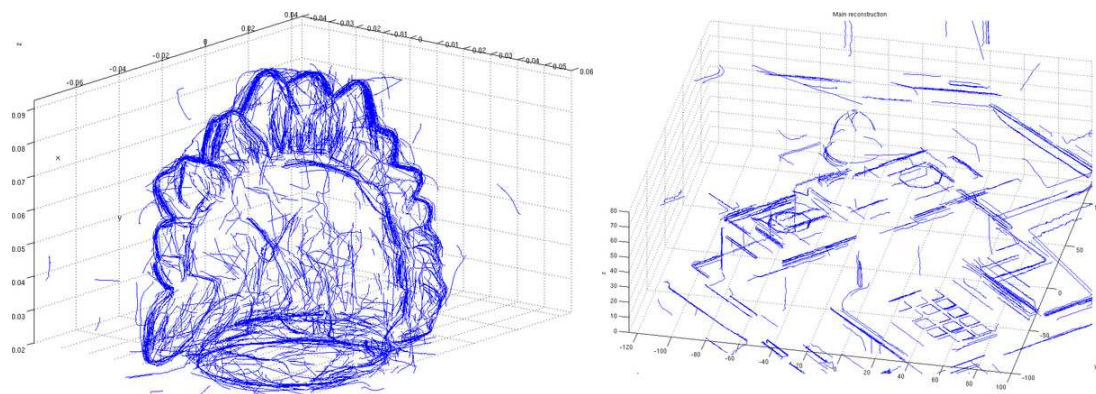


Figure 5.17: The 3D curve sketches for the Dino and Capitol sequences.

Dataset	# 3D Curves	# Hypot. Views	Min. Length τ_l	Time (s)
Dino	3712	100	20	466
Capitol	1231	30	40	567
Capitol High	1742	30	40	633
Downtown	2340	30	40	985

Table 5.2: Total running times for our research multiview curve stereo module, *including loading of edge detection and linking from disk, and computation of distance transforms*. All times are for an Intel Xeon 3.2GHz and code with multithreading disabled. The number of 3D curves column indicate how many curves were generated by running our algorithm with the specified number of hypothesis views.

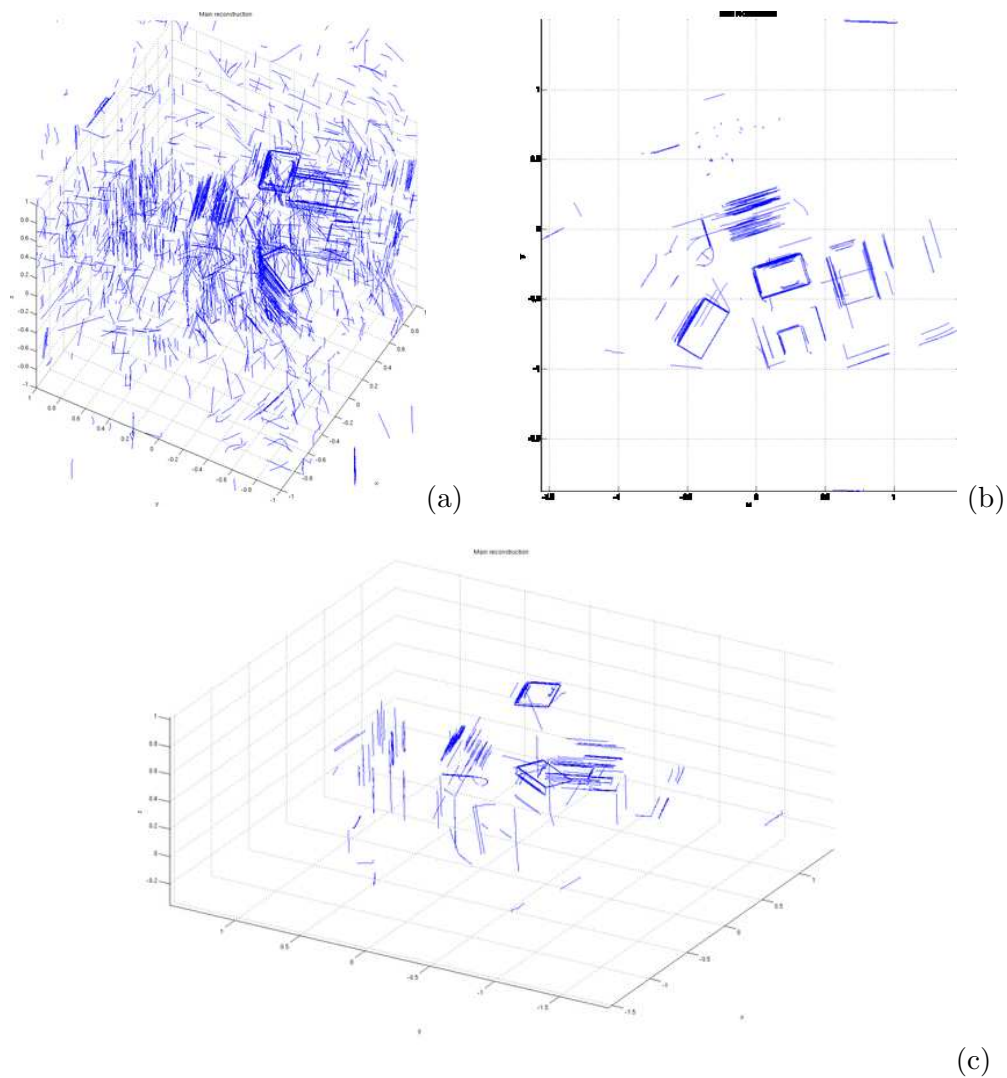


Figure 5.18: The 3D curve sketch reconstructions for the Downtown sequence using 30 sets of hypothesis views: (a) the raw 3D curve sketch showing the best candidate curves without any support confidence threshold ($\tau_t = 0$); (b-c) top and side views of the reliable curve sketch under the operating point of 30% precision and 100% recall ($\tau_t = 100$ supporting edges). The dots seen in (b) are top views of perfectly vertical curves of buildings. The representation is very efficient while still being rich enough to be used in applications such as calibration refinement and for registering new views to the scene.

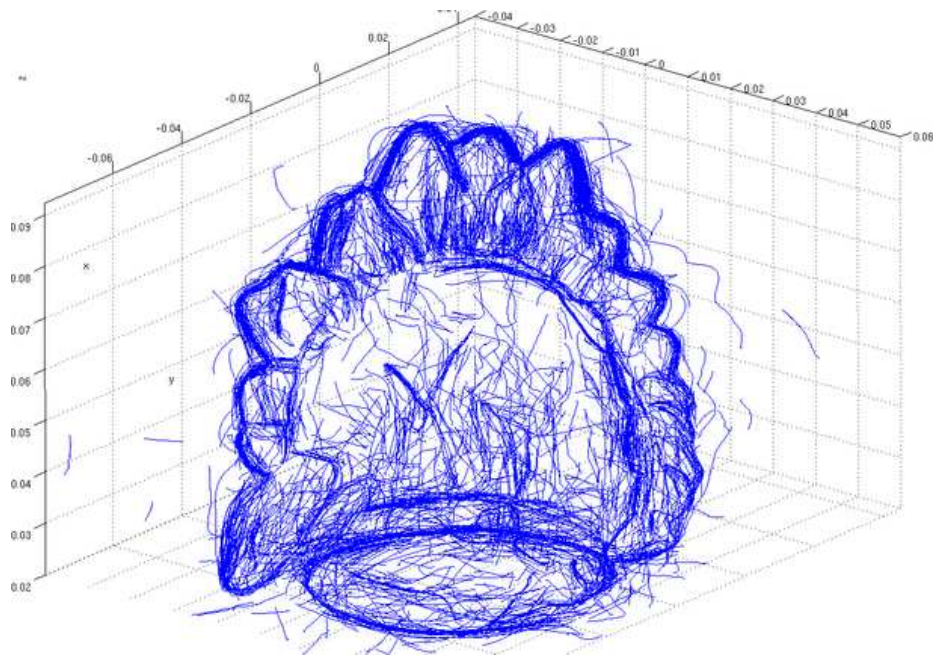


Figure 5.19: Raw 3D curve sketch reconstruction for the dinosaur with maximum recall, showing the raw output of the best candidate curves before thresholding for support. Figure 5.17, in contrast, shows only the curves at the 30% recall rate operating point.

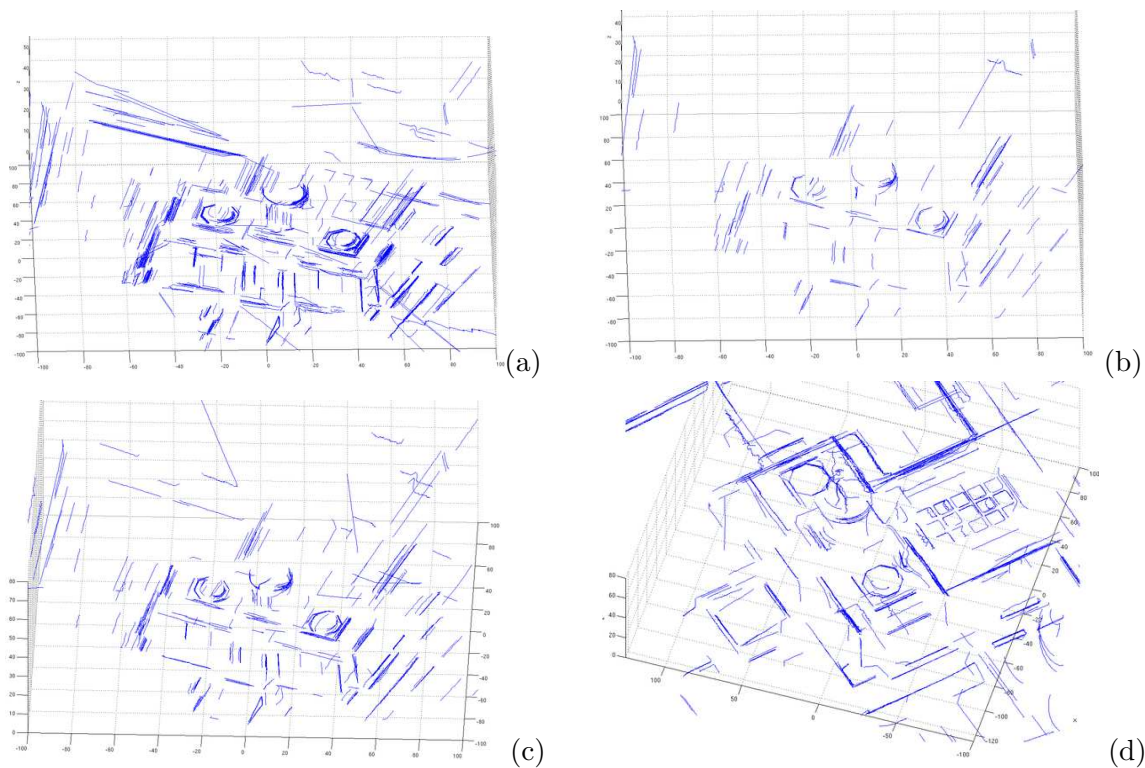


Figure 5.20: Additional 3D curve sketches for the capitol: (a) a view of the reconstruction for the maximum recall, showing all the best curves along the epipolar lines, without any threshold ($\tau_t = 0$) on the total support; (b) the same reconstruction operating at 30% recall rate and perfect precision (see Figure 8 in the original paper for the operating point parameters), being more sparse but much more reliable, used as input to curve-based calibration refinement. (d) The same reconstruction at 50% recall rate, which yields 100% precision in the evaluation, but which might contain false positives here because we run the system for many 30 hypothesis views, not including the ones used in evaluation.

Dataset	SIFT (s)	Subpixel edge detector (s)	Symbolic linker (s)
Dino	1.53	0.33	2.72
Capitol	9.85	2.25	38.91
Capitol High	9.60	2.51	46.74
Downtown	11.34	3.49	56.89

Table 5.3: **Running times of feature detection:** the added cost of using edge curves as compared to SIFT. We used David Lowe’s SIFT binary, which is heavily optimized, while the edge detector and symbolic linker are research code. All times are for an Intel Xeon 3.2GHz.

may be constructed. The effort is also underway for curve-based auto calibration based on the ideas presented here.

Additional Implementation Details

The main executable for doing multiview curve stereo is `mcs`, while the executable to refine camera pose given an initial 3D curve sketch is called `mcp`. These executables can be found in my contrib folder in `lemsvx1src`.

Multiview Curve Stereo

This module is responsible for generating a 3D curve sketch from the given video sequence and camera calibration. In order to run MCS on a sequence, we can use the `mcs` executable directly, or we can use the `mcs_rec` script.

I usually have scripts `mcs_rec_dataset_name`, where `dataset_name` is a stamp for the dataset, and I set the optimal parameters for the dataset inside such script. The `mcs_rec` executable actually runs the job in the background, and writes the commandline parameters and other identifying info such as timing inside the result directory.

The 3D curve sketch representation

The 3D curve sketch is a representation of the 3D scene by a set of 3D contours is the 3D curve sketch. In the system, each 3D curve has an associated pair of hypothesis views from which it was generated, together with a total inlier score of that curve over all views that were used in the MCS stage.

Multiview Curve Photogrammetry

This module accepts as input a 3D curve sketch representation, together with initial cameras, and outputs optimized cameras.

Chapter 6

Conclusion and Future Work

In this thesis we presented the differential-geometric theory of multiview projection and reconstruction of general curves, providing a unified framework to model fixed, occluding, and nonrigid curves under both a discrete set of views and differentiable camera motion. We also presented a practical curve-based multiview reconstruction and calibration system called the 3D curve sketch.

On the theoretical side, we studied how the differential geometry of curves behaves under perspective projection, including the effects of intrinsic parameters. For instance, we studied how the tangent, curvature, and curvature derivative of a space curve projects onto an image, and how the motion of the camera and of the curve relate to the projections. We also gave formulas for reconstructing differential geometry, given differential geometry at corresponding points measured in two views. In particular, this gives a novel result of reconstructing space curve torsion, given corresponding points, tangents, curvatures, and curvature derivatives measured in two views. We determined that there are no constraints in two views – any pair of corresponding points with attributed tangents, curvatures, and curvature derivatives are possible matches, as long as the basic point epipolar constraint is satisfied. There is, however, a constraint in three views: from two views one can transfer differential geometry onto a third and enforce measurements to match the reprojection. This has been used in the practical work in curve-based multiview stereo described in Section 5.

Preliminary results on the multiview differential geometry of surfaces was also provided in an Appendix, describing their local induced flow in images and differential aspects of shading. On another appendix we also described preliminary research on geometric and qualitative aspects of epipolar geometry estimation, which provide foundations for future work on automatic curve-based structure from motion for a large number of views.

On the practical side, we have presented a novel framework for multiview reconstruction and calibration refinement based on image curve content. The approach augments existing interest-point based and stereo approaches in providing explicit curve geometry as well in extending applications where the assumption of these methods fail but image curve content is present. A key capability is integration across many views in large-scale applications such as in using Google’s Street View

imagery for reconstructing cities and beyond.

6.1 Future Work

This thesis is part of a greater effort of augmenting Multiple View Geometry to handle general curved structures, including curves, surfaces and non-rigid phenomena. The main lines of future research are: the theory and practice of the multiview differential geometry of surfaces, and the automatic curve-based calibration of multiple views. A more detailed list of future work is given below.

6.1.1 Theory

- Further development and refinement of the differential-geometric theory on surfaces. This includes the study of shading under illumination models other than Lambertian.
- Curve-based calibration of 3 views. Research is underway in conjunction with prof. Peter Giblin to solve the problem of determining trinocular relative pose from corresponding point-tangents across 3 views. Coupled with our single-view pose calibration method from Section 4, this would allow for complete curve-based structure from motion systems starting from a set of images without any initial calibration. Extending these ideas to include curvature is another research possibility.
- Study of how differential geometry transforms under lens distortion.
- Study of the multiview differential geometry of highlights contours or specular reflection contours.
- Further study of nonrigid phenomena.

6.1.2 Practice

- Enrichment of the proposed 3D curve sketch with surface patches. The 3d curve sketch system is expected to form the initial building block in a broader effort to use image evidence of the explicit geometry of curves and surfaces and reconstruct these by integrating information across many views. The 3D curve sketch was designed to be the initial scaffold on which surfaces may be constructed. It forms a reliable structure from which to bootstrap a larger reconstruction system that works under general conditions. One idea for incorporating surfaces is the use of occlusion relationships to define a rough, preliminary surface structure between 3D curves. This initial model can then be optimized for photoconsistency to obtain a more precise surface model. One way to carry this out is to use a hypothesize-and-test framework, where we form hypotheses for whether there is a surface patch between two 3D curves, and test these hypothesis by reprojecting onto views and verifying consistency. Hypothesis formation could be image-based where we project the 3D curve sketch onto an image (or a series of

reference images), and form region fragments between neighboring image curves. Each surface hypothesis could be tested by forming a rough surface model between the underlying 3D curves of the patch (*e.g.*, a by fitting a minimal surface), and projecting the entire curve sketch onto another view. Using the model for the surface patch, one can predict what other curves should be occluded in case the hypothesis is correct. We then look in the image for any edges that would support the curves that should have been occluded. If there is support, then the curves are not actually occluded and the surface hypothesis must be false. The basic idea of using image region fragments has been successful as part of the top-performing 2-view stereo algorithms in the literature.

- Use of curvature in the 3D curve sketch reprojection cost. Currently, only tangent information is used to disambiguate stereo matches. By augmenting the cost with curvature, we expect to get a large increase in the reliability of the matching, which allows us to use less confirmation views and to reconstruct correspondences that would otherwise be deemed unreliable. We expect the addition of even a rough form of curvature will be helpful.
- Use of curvelets in the 3D curve sketch reprojection cost. This is related to the previous item. Curvelets are small local grouping hypotheses of a few edgels (say 5–10) [144]. They can be used to avoid having a reprojected curve gather support from edgels coming from different image curves, by requiring the support to be consistent with the curvelets.
- Improvement of the 3D curve sketch reconstruction by consolidating the 3D curves. The system described in this thesis may reconstruct the same curve twice or more, in case the same curve appears in more than a pair of hypothesis views. We need a way of linking the correspondences and reconstructions across different pairs of hypothesis views.
- Improvement of the 3D reconstruction in the neighborhood of epipolar tangencies. In the system described in this thesis, we use a parameter τ_α which is the minimum angle of curve segments with epipolar lines. Any curve segments/subcurves forming an epipolar angle smaller than τ_α are not reconstructed. This generates a more precise but less complete 3D reconstruction. In order to reconstruct across these epipolar gaps, it is necessary to perform some kind of interpolation using the pointwise correspondence information of the reliable segments of the curve. This can be achieved by using dynamic programming, followed by an iterative optimization procedure, to find a global pointwise correspondence between the two curve fragments. Such a technique is used in the manual CAD system of [159].
- Use of the “sweep” idea to speed up curve correspondence in the 3D curve sketch system. Robert and Faugeras [131] proposed an idea of an epipolar sweep between two views so that intersections of curve fragments with epipolar lines can be done only once. This avoids a huge redundancy because the epipolar candidates for a given curve fragment in the left image is a set of curve fragment in the right image which has huge overlap with the set of epipolar candidates for another curve fragment in the left image. By using endpoint transitions as one

sweeps the epipolar lines by increasing angle, it is possible to get a very fast algorithm for determining the epipolar candidates.

- Incorporation of occluding contours into the 3D curve sketch system. This would require views to be dense, as in a video. The system proposed in this thesis is able separate stationary curves from non-stationary curves (up to the calibration accuracy). Amongst the non-stationary curves, one needs to pick out which ones are occluding contours, and which ones are not (such as specular highlight contours). An idea for doing this is to first treat each curve as if it was an occluding contour, then test this hypothesis. First, one would track/match the given curve fragment with (at least) the next and the previous frames, since 3 views are necessary for a local surface model reconstruction by [98]. Once a local 3D surface model is obtained, one can check the photoconsistency of this 3D surface model with *other* views that see it head-on. If the surface patch is photo-consistent, then the curve triplet is classified as apparent contours. The output of the system would be our 3D curve sketch augmented with surface patches near occluding contours.
- Further study of Papadopoulos’s experiments and numerics regarding the practical use of the equation relating normal curve velocities to camera calibration described in Chapter 3. It would be of great importance to make such a useful equation practical.
- Improve convergence of bundle adjustment in the 3D curve sketch system. For certain combinations of scenes and number of curves, we observed that our proposed bundle adjustment may not converge. One possibility is to use an analytic Jacobian in the curve based bundle-adjustment, since for this thesis we used a numerical Jacobian computation.
- Image-based matching of curves in two and three views, so that these correspondences can be used to bootstrap camera calibration from curves. We have done preliminary work using SIFT descriptors attached to curves, and the matching results were very promising. The descriptors are rotated to match the tangent direction of the curve at each sample. There is room for improvement in terms of efficiency, as we are computing SIFT descriptors at different scales for each curve sample, as well as matching all of them. Perhaps a subsampling strategy should be used, or even a different strategy where the histogram bins are placed on a global grid built around the entire curve.
- Extension of the qualitative/geometric ideas of Appendix D for epipolar geometry to curves.
- Development of a better numeric method to measure curvature derivative, allowing to measure 3D torsion from image curves. Perhaps even a coarse but robust measurement of curvature derivative could be useful in practice.

Appendix A

Surface differential geometry

The purpose of this section is to summarize the concepts of differential geometry of surfaces, focused on the research topics of this thesis. It serves as a review for people that already have some knowledge of differential geometry. Another goal is to explicitly clarify confusions, motivate concepts and justify notation, so this is not a brief summary. For example, we motivate and interpret the first and second fundamental forms, since most people feel uneasy with the concepts behind these abstract entities at first. In a sentence, the need for these entities comes from the desire to study the surface intrinsically, avoiding references to the extrinsic space (\mathbb{R}^3) as much as possible.

A.1 Basics

Let us begin by defining the differential of a vector function of a vector variable (also called a *map*).

Definition 4. (From [35]) Let $\mathbf{F} : U \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a differentiable map. To each $p \in U$ we associate a linear map $dF_p : \mathbb{R}^n \rightarrow \mathbb{R}^m$, which is called the differential of \mathbf{F} at p , and is defined as follows. Let $w \in \mathbb{R}^n$ and $\gamma(-\epsilon, \epsilon) \rightarrow U$ be a differentiable curve such that $\gamma(0) = p$, $\gamma'(0) = w$. Let the curve $\mathbf{\Gamma} = \mathbf{F} \circ \gamma : (-\epsilon, \epsilon) \rightarrow \mathbb{R}^m$, then:

$$dF_p(w) := \mathbf{\Gamma}'(0). \quad (\text{A.1.1})$$

This definition does not depend on the curve γ . In simple terms, the differential of a map is a linear map which, applied to any given vector $w \in \mathbb{R}^n$, gives the directional derivative of the map along w .

We can write the matrix of dF_p relative to the canonical bases of \mathbb{R}^n and \mathbb{R}^m , which is called the *Jacobian matrix* of $\mathbf{F} = (f_1, \dots, f_m)^\top$:

$$\mathbf{J} = \text{Jac}[\mathbf{F}] = \left(\frac{\partial f_i}{\partial x_j} \right) = \begin{bmatrix} \frac{\partial \mathbf{F}}{\partial x_1} & \dots & \frac{\partial \mathbf{F}}{\partial x_n} \end{bmatrix} \quad (\text{A.1.2})$$

When $n = m$, the Jacobian matrix is square and its determinant is called the *Jacobian determinant*

or just ‘*Jacobian*’, and is denoted by

$$\det \left(\frac{\partial f_i}{\partial x_j} \right) = \frac{\partial(f_1, \dots, f_m)}{\partial(x_1, \dots, x_n)}. \quad (\text{A.1.3})$$

A regular surface \mathcal{M} is a 2-dimensional submanifold of \mathbb{R}^3 . This means that, for each point $\mathbf{\Gamma}$ of the surface, there exists a ‘chart’, or ‘parametrization’, or ‘system of (local) coordinates’, which maps a region of the plane \mathbb{R}^2 into the neighborhood (called coordinate neighborhood) of the point $\mathbf{\Gamma}$ in the surface. This mapping is a *diffeomorphism*, which means it is differentiable and invertible (the inverse also being differentiable)¹. The intuition is that each point of the surface is locally equivalent to a plane, as far as differential questions are concerned. We will denote coordinates as ξ, η , and the points in a neighborhood of $\mathbf{\Gamma}$ are parametrized by $\mathbf{\Gamma}(\xi, \eta)$. Notice that many different parametrizations may be necessary to cover all points of a regular surface.

Regular surfaces admit a tangent plane $T_p(\mathcal{M})$ at every point $p \in \mathcal{M}$, given by $d\mathbf{\Gamma}_p(\mathbb{R}^2) \subset \mathbb{R}^3$. The choice of parametrization determines the basis $\{\mathbf{\Gamma}_\xi, \mathbf{\Gamma}_\eta\}$ of the tangent plane. A vector $w \in T_p(\mathcal{M})$ has coordinates $(\xi'(0), \eta'(0))$ in such basis, where $(\xi(t), \eta(t))$ is the expression, in the parametrization $\mathbf{\Gamma}$, of a curve whose velocity vector at $t = 0$ is w . This comes from the chain rule: $\mathbf{\Gamma}'(t) = \mathbf{\Gamma}'(\xi(t), \eta(t)) = \mathbf{\Gamma}_\xi \xi' + \mathbf{\Gamma}_\eta \eta'$.

A.2 First Fundamental Form

Definition 5. (*First fundamental form*) The quadratic form I_p on $T_p(\mathcal{M})$ defined by

$$I_p(\mathbf{w}) = \mathbf{w} \cdot \mathbf{w} = \|\mathbf{w}\|^2, \quad (\text{A.2.1})$$

where \mathbf{w} is a tangent vector to \mathcal{M} , is called the first fundamental form of the regular surface \mathcal{M} .

The importance of the first fundamental form comes from the fact that, once it is given, we can treat metric questions (such as lengths, angles of tangent vectors, and areas) on a regular surface without further reference to the extrinsic space \mathbb{R}^3 .

For our purposes, the first fundamental form is equivalent to a *norm* attached to the tangent plane of the surface at each point. In practice, we will be dealing with the usual norm of \mathbb{R}^3 , but expressed in the basis $\{\mathbf{\Gamma}_\xi, \mathbf{\Gamma}_\eta\}$ of the tangent plane, thus applying to two-dimensional local coordinates (ξ, η) in such basis instead of the three extrinsic coordinates x, y, z .

The choice of a first fundamental form on the surface is also equivalent to the choice of an inner product defined on tangent vectors of the surface. Given a norm on tangent vectors $I(\mathbf{w}) = \|\mathbf{w}\|^2 = \mathbf{w} \cdot \mathbf{w}$, we can take inner products as

$$\mathbf{v} \cdot \mathbf{w} = \frac{1}{2} [\|\mathbf{v} + \mathbf{w}\|^2 - \|\mathbf{w}\|^2 - \|\mathbf{v}\|^2]. \quad (\text{A.2.2})$$

The first fundamental form has an associated matrix of coefficients E, F, G , which are dependent on a chosen basis $\{\mathbf{\Gamma}_\xi, \mathbf{\Gamma}_\eta\}$ of $T_p(\mathcal{M})$, which, in turn, is defined by a parametrization $\mathbf{\Gamma}(\xi, \eta)$ of \mathcal{M} .

¹The rigorous notion of a differentiable map defined on a surface is not important for now.

Let us obtain the coefficients of the first fundamental form. Any vector in $T_p(\mathcal{M})$ is the tangent vector $\mathbf{\Gamma}'(t)$ of some curve $\mathbf{\Gamma}(t) = \mathbf{\Gamma}(\xi(t), \eta(t))$ in \mathcal{M} . The coordinates of such vector in the basis $\{\mathbf{\Gamma}_\xi, \mathbf{\Gamma}_\eta\}$ is (ξ', η') . Thus, the first fundamental form can be written as

$$\begin{aligned} I_p(w) &= I_p(\mathbf{\Gamma}'(t)) = \mathbf{\Gamma}'(t) \cdot \mathbf{\Gamma}'(t) = (\mathbf{\Gamma}_\xi \xi' + \mathbf{\Gamma}_\eta \eta') \cdot (\mathbf{\Gamma}_\xi \xi' + \mathbf{\Gamma}_\eta \eta') \\ &= E\xi'^2 + 2F\xi'\eta' + G\eta'^2, \end{aligned} \quad (\text{A.2.3})$$

where E, F, G are given by:

$$\begin{aligned} E &= \mathbf{\Gamma}_\xi \cdot \mathbf{\Gamma}_\xi \\ F &= \mathbf{\Gamma}_\xi \cdot \mathbf{\Gamma}_\eta \\ G &= \mathbf{\Gamma}_\eta \cdot \mathbf{\Gamma}_\eta. \end{aligned} \quad (\text{A.2.4})$$

The arc length of a curve $\mathbf{\Gamma}(t)$ of \mathcal{M} is given by

$$s(t) = \int_0^t \|\mathbf{\Gamma}'(t)\| dt = \int_0^t \sqrt{I(\mathbf{\Gamma}'(t))} dt. \quad (\text{A.2.5})$$

Using coordinates, the curve $\mathbf{\Gamma}(t)$ can be written as $\mathbf{\Gamma}(\xi(t), \eta(t))$, thus

$$s(t) = \int_0^t \sqrt{E\xi'^2 + 2F\xi'\eta' + G\eta'^2} dt. \quad (\text{A.2.6})$$

Hence, given the coefficients E, F, G , we can forget about the embedding space \mathbb{R}^3 if we want to measure arclengths. This is in the same spirit as the Frenet frames for curves, where we write all entities in terms of this frame instead of the world frame, thus remaining intrinsic to the object.

First fundamental form of the sphere in spherical coordinates. The unit sphere can be parametrized by $\hat{\gamma} : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ as

$$\hat{\gamma}(\theta, \phi) = (\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta), \quad (\text{A.2.7})$$

whose first fundamental form is $E(\theta, \phi) = 1$, $F(\theta, \phi) = 0$, and $G(\theta, \phi) = \sin^2 \theta$. Thus, if \mathbf{w} is a tangent vector to the sphere at the point $\hat{\gamma}(\theta, \phi)$, given in the basis associated to $\hat{\gamma}(\theta, \phi)$ by

$$\mathbf{w} = a\hat{\gamma}_\theta + b\hat{\gamma}_\phi, \quad (\text{A.2.8})$$

then the square of the length of w is given by

$$\|\mathbf{w}\|^2 = I(\mathbf{w}) = Ea^2 + 2Fab + Gb^2 = a^2 + b^2 \sin^2 \theta. \quad (\text{A.2.9})$$

Definition 6. (Area) The area of a region R of a surface is defined as

$$A(R) = \int \int_Q \|\mathbf{\Gamma}_\xi \times \mathbf{\Gamma}_\eta\| d\xi d\eta = \int \int_Q \sqrt{EG - F^2} d\xi d\eta, \quad (\text{A.2.10})$$

where $\mathbf{\Gamma}(\xi, \eta)$ is a parametrization of the surface in consideration and $Q = \mathbf{\Gamma}^{-1}(R)$.

A.2.1 Gradients on Surfaces.

Definition 7. (From [35]) The gradient of a differentiable function defined on a surface $f : \mathcal{M} \rightarrow \mathbb{R}$ is a differentiable map $\nabla f : \mathcal{M} \rightarrow \mathbb{R}^3$, $p \in \mathcal{M} \mapsto \nabla f(p) \in T_p(\mathcal{M})$ such that

$$\nabla f(p) \cdot v = df_p(v), \quad \forall v \in T_p(\mathcal{M}). \quad (\text{A.2.11})$$

Proposition A.2.1. If E, F, G are the coefficients of I in a parametrization $\mathbf{\Gamma}(\xi, \eta)$, then ∇f on the range of $\mathbf{\Gamma}$ is given by

$$\nabla f = \frac{f_\xi G - f_\eta F}{EG - F^2} \mathbf{\Gamma}_\xi + \frac{f_\eta E - f_\xi F}{EG - F^2} \mathbf{\Gamma}_\eta. \quad (\text{A.2.12})$$

The definition of gradient on a curved surface is consistent with the usual definition of gradient in the plane. If the surface $\mathcal{M} = \mathbb{R}^2$ with coordinates x, y , we can then apply the proposition to show that

$$\nabla f = f_x \mathbf{e}_1 + f_y \mathbf{e}_2, \quad (\text{A.2.13})$$

where $\{\mathbf{e}_1, \mathbf{e}_2\}$ is the canonical basis of \mathbb{R}^2 .

Corollary A.2.2. The vector ∇f gives the direction of maximum variation of f . More precisely, letting $p \in \mathcal{M}$ be fixed and \mathbf{v} vary in the unit circle $\|\mathbf{v}\| = 1$ in $T_p(\mathcal{M})$, then $df_p(\mathbf{v})$ is maximum if and only if $\mathbf{v} = \nabla f / \|\nabla f\|$. Moreover, The vector ∇f is normal to all points of the level curve $C = \{q \in \mathcal{M}; f(q) = \text{const}\}$, wherever $\nabla f \neq 0$.

A.3 Normals, Gauss Maps, and Second Fundamental Forms

The normal vector \mathbf{N} at a point p is a vector orthogonal to the tangent plane at p . For a given parametrization $\mathbf{\Gamma}(\xi, \eta)$, we can choose the field of normals

$$\mathbf{N} = \frac{\mathbf{\Gamma}_\xi \times \mathbf{\Gamma}_\eta}{\|\mathbf{\Gamma}_\xi \times \mathbf{\Gamma}_\eta\|}. \quad (\text{A.3.1})$$

We could also choose $-\mathbf{N}$, as long as the choice is consistent (differentiable) in a neighborhood of the chosen point p .

Definition 8. The field of normals $\mathbf{N} : \mathcal{M} \rightarrow S^2$ maps each surface point onto the unit sphere S^2 of directions, and is called the *Gauss map*.

Definition 9. The differential of the Gauss map, dN , is called the *Weingarten map*, and its negative $-dN$ is called the *Shape operator*. The Weingarten map is a linear map giving the directional derivatives of the normal vector \mathbf{N} along any vector. More precisely, letting p be a point of the surface, and \mathbf{w} a vector in the tangent plane, then

$$\begin{aligned} dN_p : T_p(\mathcal{M}) &\rightarrow T_{N(p)}(S^2) \text{ or } T_p(\mathcal{M}) \\ \mathbf{w} &\mapsto dN_p(\mathbf{w}) \end{aligned} \quad (\text{A.3.2})$$

where $dN_p(\mathbf{w})$ is given by the following rule. If $\alpha(t)$ is a curve in the surface such that $\alpha(0) = p$ and $\alpha'(0) = \mathbf{w}$, then $\mathbf{N}(t) := \mathbf{N} \circ \alpha(t)$. In the parametrization $\Gamma(\xi, \eta)$, the vector \mathbf{w} has coordinates $(\xi'(t), \eta'(t))$, and $dN_p(\mathbf{w}) = dN_p(\xi'(t), \eta'(t)) = \mathbf{N}'(t)$. Since $T_p(\mathcal{M})$ and $T_{N(p)}(S^2)$ are parallel planes, we can see dN_p as a map from $T_p(\mathcal{M})$ to $T_p(\mathcal{M})$.

Proposition A.3.1. *The differential $dN_p : T_p(\mathcal{M}) \rightarrow T_p(\mathcal{M})$ of the Gauss map is a self-adjoint linear map.*

Proof. See [35, p. 140]. ■

For completeness, we include the definition of a self-adjoint linear map:

Definition 10. A linear map $A : V \rightarrow V$ is *self-adjoint* if $(A\mathbf{v}) \cdot \mathbf{w} = \mathbf{v} \cdot (A\mathbf{w})$ for vectors $\mathbf{v}, \mathbf{w} \in V$.

For an orthonormal basis, the matrix (a_{ij}) of a self-adjoint linear map A is symmetric. To each such map, we can associate a bilinear form $B(\mathbf{v}, \mathbf{w}) = (A\mathbf{v}) \cdot \mathbf{w}$ and a quadratic form $Q(\mathbf{v}) = B(\mathbf{v}, \mathbf{v})$. Self-adjoint linear maps, bilinear forms, and quadratic forms are in 1-1 correspondence with each other, therefore being equivalent.

Since the Weingarten map dN_p is self-adjoint, we can associate a quadratic form Q to it, given by $Q(\mathbf{v}) = dN_p(\mathbf{v}) \cdot \mathbf{v}$. For convenience, the literature works with the form $II_p(\mathbf{v}) = -Q(\mathbf{v})$:

Definition 11. The quadratic form II_p , defined in $T_p(\mathcal{M})$ by $II_p(\mathbf{v}) = -dN_p(\mathbf{v}) \cdot \mathbf{v}$, is called the *second fundamental form* of \mathcal{M} at p .

It can be shown that for a unit vector $\hat{\mathbf{v}}$, $II_p(\hat{\mathbf{v}}) = \kappa_n(p)$, where $\kappa_n(p)$ is the normal curvature in the direction $\hat{\mathbf{v}}$. This is the reason for the negative sign in the definition of the second fundamental form: we want this interpretation to hold without a negative sign.

There is a general theorem involving self-adjoint linear maps and the extrema of the associated quadratic form, which we will apply to define the principal curvatures. The theorem is:

Theorem A.3.2. *(From [35]) The eigenvalues of a self-adjoint linear map $A : V \rightarrow V$ are the extrema of the associated quadratic form on the unit circle of V . The corresponding unit eigenvectors form an orthonormal basis. They diagonalize the matrix A of the considered linear map, i.e., the matrix A of the linear map relative to the eigenvectors is diagonal, the diagonal elements being the eigenvalues.*

Hence, associated to $II_p(\hat{\mathbf{v}}) = \kappa_n$ there are maximum and minimum normal curvatures κ_1 and κ_2 , which are also eigenvalues of $-dN_p$.

Definition 12. *(Principal curvatures and directions)* The maximum and minimum normal curvatures κ_1 and κ_2 are called *the principal curvatures* at p , the corresponding directions (given by the corresponding eigenvalues of $-dN_p$) being called *the principal directions*.

Another useful fact of general linear maps $A : V \rightarrow V$ is that the trace and determinant of A are independent on the choice of bases for V .

Definition 13. (*Gaussian and mean curvatures*) Let dN_p be the differential of the Gauss map. Then

$$\mathcal{K} = \kappa_1 \kappa_2 = \det dN_p \quad (\text{A.3.3})$$

$$\mathcal{H} = \frac{\kappa_1 + \kappa_2}{2} = -\frac{1}{2} \text{trace } dN_p \quad (\text{A.3.4})$$

Definition 14. (*Umbilical points*) The point $p \in \mathcal{M}$ is called an *umbilical point* if $\kappa_1 = \kappa_2$, and the umbilical points having $\kappa_1 = \kappa_2 = 0$ are called *planar points*.

Definition 15. (*Asymptotic directions*) An asymptotic direction of \mathcal{M} at $p \in \mathcal{M}$ is a direction of $T_p(\mathcal{M})$ for which the normal curvature is zero. An asymptotic curve of \mathcal{M} having its tangents along asymptotic directions at all points.

Proposition A.3.3. (*Euler's formula*) The normal curvature in any direction θ measured from a principal direction corresponding to principal curvature κ_1 is given by

$$\kappa_n(\theta) = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta \quad (\text{A.3.5})$$

Proof. Let $\mathbf{w} = \hat{\rho} \hat{\mathbf{w}} = \hat{\rho}(\sin \theta, \cos \theta)$ be a parametrization for the tangent plane in the basis $\{\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2\}$, where $\bar{\mathbf{e}}_1$ and $\bar{\mathbf{e}}_2$ are along the principal directions. Then

$$II_p(\mathbf{w}) = -dN_p(\mathbf{w}) \cdot \mathbf{w} = -dN_p(\hat{\rho} \hat{\mathbf{w}}) \cdot \hat{\rho} \hat{\mathbf{w}} = -\hat{\rho}^2 dN_p(\hat{\mathbf{w}}) \cdot \hat{\mathbf{w}} \quad (\text{A.3.6})$$

$$= -\hat{\rho}^2 dN_p(\sin \theta \bar{\mathbf{e}}_1 + \cos \theta \bar{\mathbf{e}}_2) \cdot (\sin \theta \bar{\mathbf{e}}_1 + \cos \theta \bar{\mathbf{e}}_2), \quad (\text{A.3.7})$$

and, since dN_p is linear and \mathbf{e}_1 and \mathbf{e}_2 are its eigenvectors, we have

$$= \hat{\rho}^2 (\sin \theta \kappa_1 \bar{\mathbf{e}}_1 + \cos \theta \kappa_2 \bar{\mathbf{e}}_2) \cdot (\sin \theta \bar{\mathbf{e}}_1 + \cos \theta \bar{\mathbf{e}}_2) \quad (\text{A.3.8})$$

Thus, we have:

$$II_p(\mathbf{w}) = \hat{\rho}^2 (\kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta) \quad (\text{A.3.9})$$

The last expression is known as the *Euler formula*; actually, it is just the expression of the second fundamental form in the basis $\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2$. For a unit vector $\hat{\mathbf{w}}$, $\hat{\rho} = 1$, then $II_p(\hat{\mathbf{w}})$ is the normal curvature along $\hat{\mathbf{w}}$, so the Euler formula assumes the form stated in the theorem. ■

The second fundamental form can be shown to be the osculating paraboloid to the surface, *i.e.*, the second-order Taylor approximation to the surface.

Proposition A.3.4. Consider a local coordinate system such that \bar{x} and \bar{y} are along principal directions, and \bar{z} is along a chosen normal direction \mathbf{N} of the surface \mathcal{M} . Then

$$\bar{z} = \frac{1}{2} II_p(\mathbf{w}) = \frac{1}{2} \kappa_1 \bar{x}^2 + \frac{1}{2} \kappa_2 \bar{y}^2, \quad (\text{A.3.10})$$

where $\mathbf{w} = (\bar{x}, \bar{y})$ is a tangent vector in the tangent plane $T_p(\mathcal{M})$. The above equation defines the osculating paraboloid of the surface \mathcal{M} .

Proof.

$$II_p(\mathbf{w}) = -dN_p(\mathbf{w}) \cdot \mathbf{w} = -dN_p(\bar{x}\bar{\mathbf{e}}_1 + \bar{y}\bar{\mathbf{e}}_2) \cdot (\bar{x}\bar{\mathbf{e}}_1 + \bar{y}\bar{\mathbf{e}}_2), \quad (\text{A.3.11})$$

and, since dN_p is linear and $\{\mathbf{e}_1, \mathbf{e}_2\}$ its eigenvectors, we have

$$II_p(\mathbf{w}) = (\bar{x}\kappa_1\mathbf{e}_1 + \bar{y}\kappa_2\mathbf{e}_2) \cdot (\bar{x}\mathbf{e}_1 + \bar{y}\mathbf{e}_2) \quad (\text{A.3.12})$$

$$= \kappa_1\bar{x}^2 + \kappa_2\bar{y}^2 = 2\bar{z} \quad (\text{A.3.13})$$

■

Thus, the second fundamental form is our paraboloid model equation in local coordinates given by Equation (B.1.1) (multiplied by 2). The previous proposition shows that we can think about the second fundamental form as the second-order approximation to the surface, *i.e.*, the osculating paraboloid. This is very insightful for our research, but it is not emphasized in many textbooks.

Dupin Indicatrix. The Dupin indicatrix can be defined as a level curve of the osculating paraboloid. Its equation is given by:

$$II_p(\mathbf{w}) = \pm\epsilon, \quad (\text{A.3.14})$$

or, in local coordinates \bar{x}, \bar{y} , as

$$II_p(\bar{x}, \bar{y}) = \kappa_1\bar{x}^2 + \kappa_2\bar{y}^2 = \pm\epsilon, \quad (\text{A.3.15})$$

where ϵ is usually fixed at 1. The Dupin indicatrix is shown to be approximately equal to the intersection of the surface \mathcal{M} with a plane parallel to $T_p(S)$ having distance 2ϵ from p (to first order, in the limit $\epsilon \rightarrow 0$). Since the Dupin indicatrix is a level curve of the osculating paraboloid II_p , it assumes the shape of an ellipse whenever II_p is an elliptic paraboloid, and the shape of a hyperbola whenever II_p is a hyperbolic paraboloid. We say that, in those cases, p is called an *elliptic point* and a *hyperbolic point*, respectively.

A.3.1 Conjugacy

Given the importance of conjugacy for our research, *e.g.*, in relating occluding contours to viewing direction, we will give a more comprehensive treatment to this topic.

Definition 16. Let p be a point of a surface \mathcal{M} . Two nonzero vectors $\mathbf{w}_1, \mathbf{w}_2 \in T_p(\mathcal{M})$ are conjugate if $dN_p(\mathbf{w}_1) \cdot \mathbf{w}_2 = \mathbf{w}_1 \cdot dN_p(\mathbf{w}_2) = 0$. Two directions r_1, r_2 at p are conjugate if there is a pair of conjugate vectors along them.

Since the condition for a conjugate point does not depend on the scale of the two vectors, it is a projective relationship, *i.e.*, it is invariant to projective transformations. Koenderink's book [89, p.230] also talks about conjugacy as a "generalization" of orthogonality in affine and projective spaces. In order to have an intuition of this, let us first give a geometric construction of conjugate

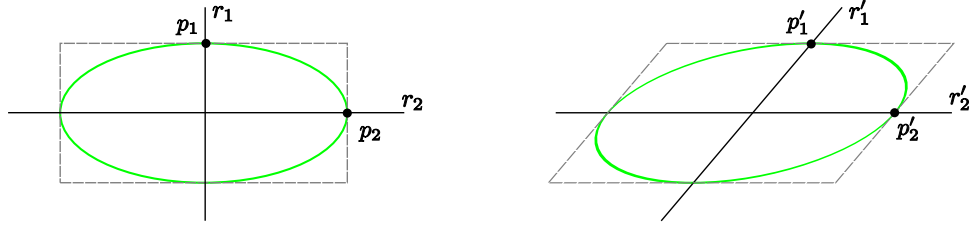


Figure A.1: Orthogonal (thus conjugate) directions r_1 and r_2 transform to conjugate directions r'_1 and r'_2 after an affine or projective transformation.

directions, as illustrated in Figure A.1. Consider an ellipse or any other central conic curve, and consider two directions r_1 and r_2 passing through the center of the curve. Let p_1 be an intersection of r_1 with the conic, and let p_2 be an intersection of r_2 with the conic. Two directions r_1 and r_2 are conjugate with respect to a conic if the tangent to the conic at p_1 is parallel to r_2 , and vice versa: the tangent to the conic at p_2 is parallel to r_1 . In the case of surfaces, the Dupin indicatrix serves as the conic in this construction. Another way of stating this construction is that a position vector of a conic is conjugate to the tangent vector at this position.

Hartley's book [69] also mentions that conjugacy with respect to the image of the absolute conic (IAC) has to do with orthogonality.

A.3.2 Gauss Map in Local Coordinates

Considering an arbitrary curve $\Gamma(t) = \Gamma(\xi(t), \eta(t)) \in \mathcal{M}$, whose tangent vector is along any desired direction in the tangent plane, we have:

$$II(\Gamma') = -d\mathbf{N}(\Gamma') \cdot \Gamma' = -(\mathbf{N}_\xi \xi' + \mathbf{N}_\eta \eta') \cdot (\Gamma_\xi \xi' + \Gamma_\eta \eta') \quad (\text{A.3.16})$$

$$= e\xi'^2 + 2f\xi'\eta' + g\eta'^2, \quad (\text{A.3.17})$$

where

$$e = \mathbf{N}_\xi \cdot \Gamma_\xi = \mathbf{N} \cdot \Gamma_{\xi\xi} \quad (\text{A.3.18})$$

$$f = \mathbf{N}_\xi \cdot \Gamma_\eta = \mathbf{N} \cdot \Gamma_{\xi\eta} \quad (\text{A.3.19})$$

$$g = \mathbf{N}_\eta \cdot \Gamma_\eta = \mathbf{N} \cdot \Gamma_{\eta\eta}. \quad (\text{A.3.20})$$

The matrix II of II in the basis $\{\Gamma_\xi, \Gamma_\eta\}$ is:

$$II = \begin{bmatrix} e & f \\ f & g \end{bmatrix}, \quad (\text{A.3.21})$$

and the matrix I of the first fundamental form is

$$I = \begin{bmatrix} E & F \\ F & G \end{bmatrix}. \quad (\text{A.3.22})$$

The matrix of the linear map $d\mathbf{N}$ is called the Jacobian matrix of \mathbf{N} , and can be shown to be given by:

$$\mathbf{J}[\mathbf{N}] = -\mathbf{I}^{-1}\mathbf{II}, \quad (\text{A.3.23})$$

with

$$\mathbf{I}^{-1} = \frac{1}{EG - F^2} \begin{bmatrix} G & -F \\ -F & E \end{bmatrix}, \quad (\text{A.3.24})$$

thus

$$\mathbf{J}[\mathbf{N}] = - \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} e & f \\ f & g \end{bmatrix} \quad (\text{A.3.25})$$

$$= - \frac{1}{EG - F^2} \begin{bmatrix} G & -F \\ -F & E \end{bmatrix}^{-1} \begin{bmatrix} e & f \\ f & g \end{bmatrix}. \quad (\text{A.3.26})$$

We can then write the so-called *equations of Weingarten*, which are the 4 entries (a_{ij}) of the Jacobian matrix of the Gauss map with respect to the fundamental forms:

$$a_{11} = \frac{fF - eG}{EG - F^2}, \quad (\text{A.3.27})$$

$$a_{12} = \frac{gF - fG}{EG - F^2}, \quad (\text{A.3.28})$$

$$a_{21} = \frac{eF - fE}{EG - F^2}, \quad (\text{A.3.29})$$

$$a_{22} = \frac{fF - gE}{EG - F^2}, \quad (\text{A.3.30})$$

where the first index denotes the row, and the second index the columns. Formulas for the Gaussian and mean curvatures for a given parametrization are given below:

$$\mathcal{K} = \frac{eg - f^2}{EG - F^2} \quad (\text{A.3.31})$$

$$\mathcal{H} = \frac{1}{2} \frac{eG - 2fF - gE}{EG - F^2}. \quad (\text{A.3.32})$$

The first equation comes by taking the determinant of (A.3.23). The equation for the mean curvature is obtained by writing the characteristic equation for the eigenvalue condition $dN(\eta) = -\kappa\eta$, giving

$$\kappa^2 + \kappa(a_{11} + a_{22}) + a_{11}a_{22} - a_{12}a_{21} = 0 \quad (\text{A.3.33})$$

or, from the elementary formulas of sum and product of the roots of a quadratic equation,

$$\kappa^2 - 2\mathcal{H}\kappa + \mathcal{K} = 0 \quad (\text{A.3.34})$$

From this, principal curvatures are:

$$\kappa = \mathcal{H} \pm \sqrt{\mathcal{H}^2 - \mathcal{K}}. \quad (\text{A.3.35})$$

Observe that this is not valid at umbilical points, where $\mathcal{H}^2 = \mathcal{K}$.

We can classify the surface point p according to the Gaussian curvature and the principal curvatures, motivated, among other things, by the shape of the Dupin indicatrix at p :

Definition 17. A point p of a surface \mathcal{M} is called

1. *Elliptic* if: $\mathcal{K} = \det(dN_p) > 0 \iff \kappa_1\kappa_2 > 0 \iff II_p(w) = \pm\epsilon$ is an ellipse $\iff eg - f^2 > 0$.
2. *Hyperbolic* if: $\mathcal{K} = \det(dN_p) < 0 \iff \kappa_1\kappa_2 < 0 \iff II_p(w) = \pm\epsilon$ is a hyperbola $\iff eg - f^2 < 0$.
3. *Parabolic* if: $\mathcal{K} = \det(dN_p) = 0$ and $dN_p \neq 0 \iff$ only one of κ_1 or κ_2 are zero (there is no simple characterization in terms of the Dupin indicatrix).
4. *Planar* if $dN_p = 0$ (thus $\mathcal{K} = 0$) $\iff \kappa_1 = \kappa_2 = 0$.

An interpretation of the Gaussian curvature is given by

$$K = \lim_{A \rightarrow 0} \frac{A'}{A}, \quad (\text{A.3.36})$$

where A is the area of a region B around the considered surface point, and A' is the area of the image of B by the Gauss map $N : S \rightarrow S^2$.

A.3.3 Surface as the graph of a function $f(x, y)$

Theorem A.3.5. *The Eigenvalues of the Weingarten Matrix of $z = f(x, y)$ are, for $\mathcal{H} \geq 0$*

$$\begin{cases} \kappa_1 = \mathcal{H} + \sqrt{\mathcal{H}^2 - K} \\ \kappa_2 = \mathcal{H} - \sqrt{\mathcal{H}^2 - K}, \end{cases} \quad (\text{A.3.37})$$

and for $\mathcal{H} < 0$

$$\begin{cases} \kappa_1 = \mathcal{H} - \sqrt{\mathcal{H}^2 - K} \\ \kappa_2 = \mathcal{H} + \sqrt{\mathcal{H}^2 - K}, \end{cases} \quad (\text{A.3.38})$$

where \mathcal{H} and K are defined as

$$\begin{cases} \mathcal{H} = \frac{(f_{xx} + f_{yy}) + (f_y^2 f_{xx} - 2f_x f_y f_{xy} + f_x^2 f_{yy})}{2(1 + f_x^2 + f_y^2)^{\frac{3}{2}}} \\ K = \frac{2(f_{xx} f_{yy} - f_{xy}^2)}{(1 + f_x^2 + f_y^2)^2}. \end{cases} \quad (\text{A.3.39})$$

Proof. When the surface is expressed as a graph $z = f(x, y)$, we can use the standard differential geometry of surfaces [35] described in the previous section to express curvatures explicitly as derivatives of the function. In general, the principal curvatures κ_1 and κ_2 and the principal directions \bar{e}_1

and \bar{e}_2 are on the eigenvalues and eigenvectors, respectively, of the Weingarten Matrix which is the product of the inverse of the first fundamental form and the second fundamental form:

$$W = - \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \begin{bmatrix} e & f \\ f & g \end{bmatrix}. \quad (\text{A.3.40})$$

For the specific case of a graph $(x, y, f(x, y))$, the tangent plane to the surface is spanned by $(1, 0, f_x)$ and $(0, 1, f_y)$, with normal $\frac{(-f_x, -f_y, 1)}{\sqrt{1+f_x^2+f_y^2}}$, giving

$$\begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} 1 + f_x^2 & f_x f_y \\ f_x f_y & 1 + f_y^2 \end{bmatrix}, \quad (\text{A.3.41})$$

$$\begin{bmatrix} e & f \\ f & g \end{bmatrix} = \frac{1}{\sqrt{1 + f_x^2 + f_y^2}} \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}. \quad (\text{A.3.42})$$

Thus, given that $EG - F^2 = 1 + f_x^2 + f_y^2$, we have

$$\begin{aligned} W &= \frac{1}{(1 + f_x^2 + f_y^2)^{\frac{3}{2}}} \begin{bmatrix} 1 + f_y^2 & -f_x f_y \\ -f_x f_y & 1 + f_x^2 \end{bmatrix} \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}, \\ &= \frac{1}{(1 + f_x^2 + f_y^2)^{\frac{3}{2}}} \begin{bmatrix} (1 + f_y^2)f_{xx} - f_x f_y f_{xy} & (1 + f_y^2)f_{xy} - f_x f_y f_{yy} \\ -f_x f_y f_{xx} + (1 + f_x^2)f_{xy} & -f_x f_y f_{xy} + (1 + f_x^2)f_{yy} \end{bmatrix}. \end{aligned} \quad (\text{A.3.43})$$

The eigenvalues of a matrix of the form $\frac{1}{e} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ satisfy the quadratic equation $e^2 \lambda^2 - e(a+d)\lambda + (ad - bc) = 0$, which in this case gives

$$\begin{aligned} (1 + f_x^2 + f_y^2)^3 \lambda^2 &- (1 + f_x^2 + f_y^2)^{\frac{3}{2}} [(f_{xx} + f_{yy}) + f_y^2 f_{xx} - 2f_x f_y f_{xy} + f_x^2 f_{yy}] \lambda \\ &+ [((1 + f_y^2)f_{xx} - f_x f_y f_{xy})(1 + f_x^2)f_{yy} - f_x f_y f_{xy} - \\ &((1 + f_y^2)f_{xy} - f_x f_y f_{yy})(1 + f_x^2)f_{xy} - f_x f_y f_{xx}] = 0. \end{aligned} \quad (\text{A.3.44})$$

The last term can be simplified as follows

$$\begin{aligned} ad - bc &= (1 + f_x^2)(1 + f_y^2)f_{xx}f_{yy} + f_x^2 f_y^2 f_{xy}^2 - (1 + f_x^2)(1 + f_y^2)f_{xy}^2 - f_x^2 f_y^2 f_{xx}f_{yy} \\ &= (1 + f_x^2)(1 + f_y^2)(f_{xx}f_{yy} - f_{xy}^2) - f_x^2 f_y^2 (f_{xx}f_{yy} - f_{xy}^2) \\ &= (1 + f_x^2 + f_y^2)(f_{xx}f_{yy} - f_{xy}^2). \end{aligned}$$

The quadratic Equation A.3.44 can now be simplified as

$$\begin{aligned} (1 + f_x^2 + f_y^2)^2 \lambda^2 &- (1 + f_x^2 + f_y^2)^{\frac{1}{2}} [(f_{xx} + f_{yy}) + (f_y^2 f_{xx} - 2f_x f_y f_{xy} \\ &+ f_x^2 f_{yy})] \lambda + (f_{xx}f_{yy} - f_{xy}^2) = 0, \end{aligned} \quad (\text{A.3.45})$$

where the solutions for λ are the principal curvatures κ_1 and κ_2 . The mean curvature \mathcal{H} , half the sum of the two eigenvalues $\mathcal{H} = (\kappa_1 + \kappa_2)/2$, can be written as

$$\mathcal{H} = \frac{(f_{xx} + f_{yy}) + (f_y^2 f_{xx} - 2f_x f_y f_{xy} + f_x^2 f_{yy})}{2(1 + f_x^2 + f_y^2)^{\frac{3}{2}}}. \quad (\text{A.3.46})$$

Similarly, the Gaussian curvature \mathcal{K} , the product of the two eigenvalues $\mathcal{K} = \kappa_1 \kappa_2$, can be written as

$$\mathcal{K} = \frac{2(f_{xx} f_{yy} - f_{xy}^2)}{(1 + f_x^2 + f_y^2)^2}. \quad (\text{A.3.47})$$

We can then write from

$$\begin{cases} \frac{\kappa_1 + \kappa_2}{2} = \mathcal{H} \\ \kappa_1 \kappa_2 = \mathcal{K}, \end{cases} \quad (\text{A.3.48})$$

a single equation $\kappa_1(2S - \kappa_1) = K$ or

$$\kappa_1^2 - 2S\kappa_1 + K = 0, \quad (\text{A.3.49})$$

which gives the result. The Eigenvectors can then be expressed as

$$\begin{bmatrix} -b \\ a - e\lambda \end{bmatrix} = \begin{bmatrix} f_x f_y f_{yy} - (1 + f_x^2) f_{xy} \\ (1 + f_y^2) f_{xx} - f_x f_y f_{xy} - (1 + f_x^2 + f_y^2)^{3/2} \kappa_1 \end{bmatrix} \quad (\text{A.3.50})$$

■

A.4 Useful identities for reference

Vector product identities

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c} \quad (\text{A.4.1})$$

Two other identities follow from the above one:

$$(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c}) \quad (\text{A.4.2})$$

$$(\mathbf{a} \times \mathbf{b}) \times (\mathbf{c} \times \mathbf{d}) = [\mathbf{abd}]\mathbf{c} - [\mathbf{abc}]\mathbf{d}, \quad (\text{A.4.3})$$

where the latter expression indicates that for the intersection of two planes, the common vector \mathbf{v} is in both planes.

$$\|\mathbf{a} \times \mathbf{b}\|^2 + (\mathbf{a} \cdot \mathbf{b})^2 = \|\mathbf{a}\|^2 \cdot \|\mathbf{b}\|^2 \quad (\text{A.4.4})$$

If A is a linear map, then

$$A\mathbf{a} \times A\mathbf{b} = \det A(\mathbf{a} \times \mathbf{b}) \quad (\text{A.4.5})$$

Appendix B

Multiview Differential Geometry of Surfaces

In this Appendix we present a preliminary study and theoretical results on the local differential geometry of surfaces in multiple views, including surface-induced image flow and the multiview behavior of shading. Appendix A reviews useful background material related to the differential geometry of surfaces.

B.1 Notation

The object local coordinate system is defined to be centered at a select point on the surface of the object, $\mathbf{\Gamma}_c(0) = \rho_c(0) [\xi_c(0), \eta_c(0), 1]^\top$, and with the direction \bar{z} aligned with the surface normal and with \bar{x} and \bar{y} defined as the direction of principal curvatures κ_1 and κ_2 , assuming they are different from each other (non-umbilical points), and such that the resulting coordinate system has same orientation as the camera coordinate system.

A point in local coordinates is hence denoted $\bar{\mathbf{\Gamma}} = [\bar{x}, \bar{y}, \bar{z}]^\top$. The object surface near the origin $\bar{\mathbf{\Gamma}}_c = [0, 0, 0]^\top$ is then described in Monge patch form as:

$$\bar{z} = \frac{1}{2}\kappa_1\bar{x}^2 + \frac{1}{2}\kappa_2\bar{y}^2 + \dots \quad (\text{B.1.1})$$

We define the curvature matrix \mathbf{K} as:

$$\mathbf{K} = \begin{bmatrix} -\kappa_1 & 0 & 0 \\ 0 & -\kappa_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (\text{B.1.2})$$

which is the canonical matrix for a paraboloid, and write (B.1.1) as

$$e_3^\top \bar{\mathbf{\Gamma}} + \frac{1}{2} \bar{\mathbf{\Gamma}}^\top \mathbf{K} \bar{\mathbf{\Gamma}} + \dots = 0 \quad (\text{B.1.3})$$

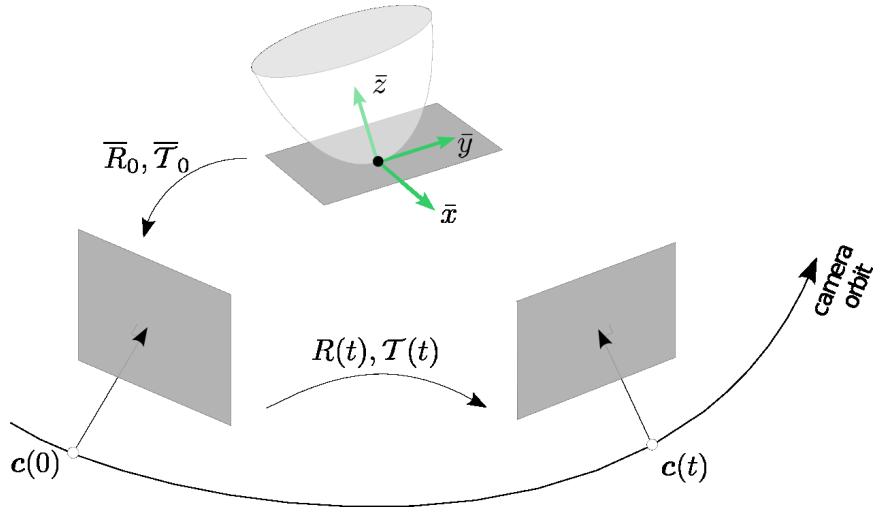


Figure B.1: Projection of a quadric in images.

where we use the notation

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (\text{B.1.4})$$

Later in this chapter we assume the object can locally be approximated by truncating the above equation to second-order, obtaining a special quadric surface model \mathcal{M} . This second-order surface approximation is an elliptic paraboloid if $\kappa_1\kappa_2 > 0$, a hyperbolic paraboloid if $\kappa_1\kappa_2 < 0$, or a plane if $\kappa_1 = \kappa_2 = 0$. The 3D points written in the camera coordinates at time t can be written in terms of the Monge patch coordinates as in Figure B.1,

$$\mathbf{\Gamma}(t) = \bar{R}(t)\bar{\mathbf{\Gamma}}(t) + \bar{\mathcal{T}}(t) \quad (\text{B.1.5})$$

where

$$\bar{R}(t) = R(t)\bar{R}_0 \quad (\text{B.1.6})$$

$$\bar{\mathcal{T}}(t) = R(t)\bar{\mathcal{T}}_0 + \mathcal{T}(t) \quad (\text{B.1.7})$$

We also have the following notation:

$$\bar{R}_0 = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \quad \bar{\mathcal{T}}(t) = \begin{bmatrix} \bar{\mathcal{T}}_x \\ \bar{\mathcal{T}}_y \\ \bar{\mathcal{T}}_z \end{bmatrix}. \quad (\text{B.1.8})$$

The unit normal $\mathbf{N} = (\mathbf{N}_x, \mathbf{N}_y, \mathbf{N}_z)$ to the surface, written in camera coordinates, is determined by the slant and tilt angles (more later). The world coordinates for the normal will be denoted \mathbf{N}^w , and we can write $\mathbf{N} = R\mathbf{N}^w$.

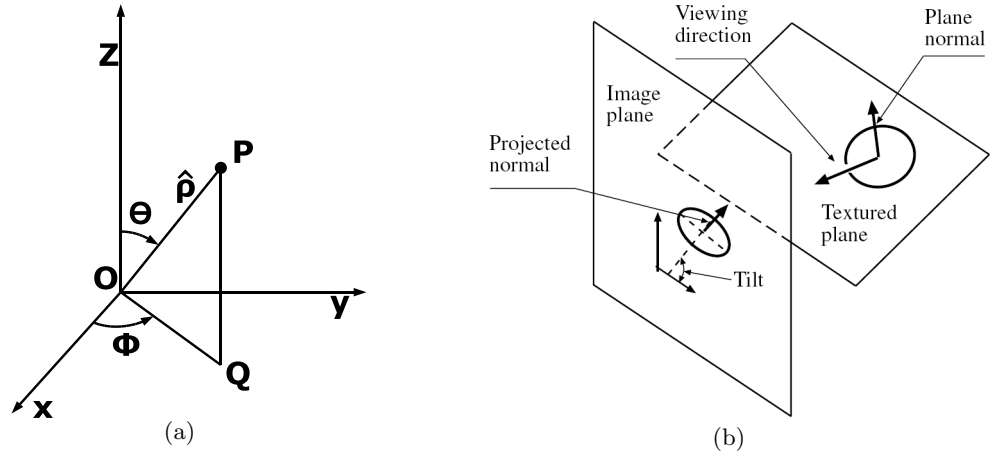


Figure B.2: (a) Spherical coordinates used in this text, and (b) The orientation of a (tangent) plane with respect to the camera plane can be given by the *slant* – which is the angle between the normal of the plane and the viewing direction – and the *tilt* – which is the angle the projected normal makes with the x -axis of the camera coordinate system (from [55]).

The third column of the rotation matrix $\bar{R}(t)$ is the normal \mathbf{N} written in camera coordinates at time t :

$$\mathbf{N}(t) = \bar{R}(t)\mathbf{e}_3 \quad (\text{B.1.9})$$

where \mathbf{e}_3 is the normal in local coordinates of the surface patch. Therefore, only two angles, *e.g.* slant σ and tilt ν , appear in the third column. The first two columns are the camera coordinates of unit vectors along the \bar{x} , and \bar{y} directions, and this is completely defined by the normal and an in-plane rotation ψ . This is given in more details as follows. The surface normal may be described in terms of slant σ and tilt ν ,

$$\mathbf{N} = (\cos \nu \sin \sigma, \sin \nu \sin \sigma, \cos \sigma)^\top, \quad (\text{B.1.10})$$

see Figure B.2(b).

We define ψ as the in-plane rotation of the local x and y axes with respect to a predefined reference coordinate frame of the tangent plane. We now derive an explicit expression for this reference coordinate frame in terms of the camera frame, with the intention of deriving a parametrization of the rotation matrix by the angles σ, ν, ψ . The reference frame for measuring ψ , denoted $\mathbf{e}_1^{\mathbf{N}}, \mathbf{e}_2^{\mathbf{N}}$, is conveniently defined with respect to the camera frame: $\mathbf{e}_1^{\mathbf{N}}$ is the orthogonal projection of \mathbf{e}_1 onto the tangent plane with normal \mathbf{N} , and $\mathbf{e}_2^{\mathbf{N}}$ is chosen to form a right-handed frame $(\mathbf{e}_1^{\mathbf{N}}, \mathbf{e}_2^{\mathbf{N}}, \mathbf{N})$. This can be written analytically as:

$$\begin{cases} \mathbf{e}_2^{\mathbf{N}} \doteq \mathbf{N} \times \mathbf{e}_1 \\ \mathbf{e}_1^{\mathbf{N}} \doteq \mathbf{e}_2^{\mathbf{N}} \times \mathbf{N}, \end{cases} \quad (\text{B.1.11})$$

which allows us to express the local x and y direction vectors \bar{e}_1 and \bar{e}_2 , respectively, as:

$$\begin{cases} \bar{e}_1 = \cos \psi \mathbf{e}_1^{\mathbf{N}} + \sin \psi \mathbf{e}_2^{\mathbf{N}} + 0 \mathbf{N} \\ \bar{e}_2 = \sin \psi \mathbf{e}_1^{\mathbf{N}} - \cos \psi \mathbf{e}_2^{\mathbf{N}} + 0 \mathbf{N}. \end{cases} \quad (\text{B.1.12})$$

Using (B.1.10) into (B.1.11), we can write the reference vectors in terms of σ and ν

$$\begin{cases} \mathbf{e}_2^{\mathbf{N}} = (0, \cos \sigma, -\sin \nu \sin \sigma)^\top \\ \mathbf{e}_1^{\mathbf{N}} = (-\sin^2 \nu \sin^2 \sigma - \cos^2 \sigma, \sin \nu \sin \sigma, -\cos \sigma)^\top, \end{cases} \quad (\text{B.1.13})$$

which we plug into (B.1.12) to yield the principal directions in terms of σ , ν , and ψ :

$$\bar{e}_1 = \begin{bmatrix} -\cos \psi \sin^2 \nu \sin^2 \sigma - \cos \psi \cos^2 \sigma \\ \sin \psi \sin \nu \sin \sigma - \cos \psi \cos \sigma \\ -\cos \psi \cos \sigma - \sin \psi \sin \nu \sin \sigma \end{bmatrix}, \quad \bar{e}_2 = \begin{bmatrix} -\sin \psi \sin^2 \nu \sin^2 \sigma - \sin \psi \cos^2 \sigma \\ \sin \psi \sin \nu \sin \sigma - \cos \psi \cos \sigma \\ -\sin \psi \cos \sigma + \cos \psi \sin \nu \sin \sigma \end{bmatrix}, \quad (\text{B.1.14})$$

written in camera coordinates. Then, a point $\Gamma(0)$ is related to $\bar{\Gamma}$ as

$$\Gamma(0) = \bar{R}_0 \bar{\Gamma} + \bar{T}_0, \quad (\text{B.1.15})$$

where

$$\bar{R}_0 = \begin{bmatrix} -\cos \psi \sin^2 \nu \sin^2 \sigma - \cos \psi \cos^2 \sigma & -\sin \psi \sin^2 \nu \sin^2 \sigma - \sin \psi \cos^2 \sigma & \cos \nu \sin \sigma \\ \sin \psi \sin \nu \sin \sigma - \cos \psi \cos \sigma & \sin \psi \sin \nu \sin \sigma - \cos \psi \cos \sigma & \sin \nu \sin \sigma \\ -\cos \psi \cos \sigma - \sin \psi \sin \nu \sin \sigma & -\sin \psi \cos \sigma + \cos \psi \sin \nu \sin \sigma & \cos \sigma \end{bmatrix}, \quad (\text{B.1.16})$$

and $\bar{T}_0 = (\bar{T}_x, \bar{T}_y, \bar{T}_z)^\top$. Note that the first two columns of the camera matrix are the principal direction vectors written in camera coordinates and the third column is the normal in camera coordinates, all parametrized by σ , ν and ψ .

Proposition B.1.1. *Given a paraboloid $\frac{1}{2} \bar{\Gamma}^\top \mathbf{K} \bar{\Gamma} + e_3^\top \bar{\Gamma} = 0$ in the coordinate system defined by $\Gamma_0 = \bar{R}_0 \bar{\Gamma}_0 + \bar{T}$, and a camera motion defined by $R(t)$ and $\mathcal{T}(t)$. Then, the equation of the paraboloid in camera coordinates is given by:*

$$\Gamma^\top \mathbf{K}_c \Gamma - 2 \bar{T}^\top \mathbf{K}_c \Gamma + e_3^\top \bar{R}^\top \Gamma + \bar{T}^\top \mathbf{K}_c \bar{T} - e_3^\top \bar{R}^\top \bar{T} = 0. \quad (\text{B.1.17})$$

where $\mathbf{K}_c = \frac{1}{2} \bar{R} \mathbf{K} \bar{R}^\top$, \mathbf{K} is the principal curvature matrix of the surface as defined in Equation (3.6.8), $\bar{R} = R(t) \bar{R}_0$, and $\bar{T} = R(t) \bar{T}_0 + \mathcal{T}(t)$.

Proof. The main equations are:

$$\begin{cases} \frac{1}{2} \bar{\Gamma}^\top \mathbf{K} \bar{\Gamma} + e_3^\top \bar{\Gamma} = 0 & \text{(second-order surface model)} \end{cases} \quad (\text{B.1.18})$$

$$\begin{cases} \Gamma_0 = \bar{R}_0 \bar{\Gamma} + \bar{T}_0 & \text{(Monge to camera coordinates at } t = 0) \end{cases} \quad (\text{B.1.19})$$

$$\begin{cases} \Gamma(t) = R(t) \Gamma_0 + \mathcal{T}(t) & \text{(camera coordinates at time } t) \end{cases} \quad (\text{B.1.20})$$

$$\begin{cases} \Gamma(t) = \rho(t) \gamma(t) & \text{(projection equation)} \end{cases} \quad (\text{B.1.21})$$

We have from (B.1.19) and (B.1.20)

$$\begin{aligned}\mathbf{\Gamma}(t) &= R(t)\bar{R}_0\bar{\mathbf{\Gamma}} + R(t)\bar{\mathcal{T}}_0 + \mathcal{T}(t) \\ &= \bar{R}\mathbf{\Gamma} + \bar{\mathcal{T}},\end{aligned}\tag{B.1.22}$$

giving

$$\bar{\mathbf{\Gamma}} = \bar{R}(t)^\top \mathbf{\Gamma}(t) - \bar{R}(t)^\top \bar{\mathcal{T}}(t).\tag{B.1.23}$$

Substituting this into the above equation (B.1.18) (for simplicity, the dependency on t will not be written), we have

$$\frac{1}{2} \left[\bar{R}^\top \mathbf{\Gamma} - \bar{R}^\top \bar{\mathcal{T}} \right]^\top \mathbf{K} \left[\bar{R}^\top \mathbf{\Gamma} - \bar{R}^\top \bar{\mathcal{T}} \right] + e_3^\top \left[\bar{R}^\top \mathbf{\Gamma} - \bar{R}^\top \bar{\mathcal{T}} \right] = 0,\tag{B.1.24}$$

or

$$\frac{1}{2} \left[\mathbf{\Gamma} - \bar{\mathcal{T}} \right]^\top \bar{R}\mathbf{K}\bar{R}^\top \left[\mathbf{\Gamma} - \bar{\mathcal{T}} \right] + e_3^\top \bar{R}^\top \left[\mathbf{\Gamma} - \bar{\mathcal{T}} \right] = 0.\tag{B.1.25}$$

Using $\mathbf{K}_c := \frac{1}{2}\bar{R}\mathbf{K}\bar{R}^\top$, distributing, and rearranging, we get the final equation. \blacksquare

We are interested in an image neighborhood around a given center point $\gamma_c(0) = [\xi_c(0), \eta_c(0), 1]^\top$ at $t = 0$. The paraboloid approximation would then be around the 3D point $\mathbf{\Gamma}_c(0) = [x_c(0), y_c(0), z_c(0)]^\top$, with the notation $\mathbf{\Gamma}_c(t) := \rho_c(t)\gamma_c(t)$. In local coordinates, $\mathbf{\Gamma}_c(0)$ corresponds to $\bar{\mathbf{\Gamma}} = [0, 0, 0]^\top$, which, when plugged into (B.1.5), yields:

$$\mathbf{\Gamma}_c(0) = \bar{\mathcal{T}}(0) = \bar{\mathcal{T}}_0\tag{B.1.26}$$

It is then clear that $\bar{\mathcal{T}}(t)$ is the coordinates of the origin of the Monge patch with respect to the camera coordinates at time t . From (B.1.7), we can write:

$$\bar{\mathcal{T}}(t) = R(t)\bar{\mathbf{\Gamma}}_c(0) + \mathcal{T}(t)\tag{B.1.27}$$

$$\bar{\mathcal{T}}(t) = R(t)\rho_c(0)\gamma_c(0) + \mathcal{T}(t)\tag{B.1.28}$$

Note that $\bar{\mathcal{T}}(t) \neq \mathbf{\Gamma}_c(t)$, because $\mathbf{\Gamma}_c$ was defined with respect to the center of the image patch, γ_c , which for any $t \neq 0$ will not correspond to the origin of the 3D monge patch. Thus, given the image point $\gamma_c(0)$ at $t = 0$, the translation to the Monge patch is determined from a single number $\rho_c(0)$, so in total we need 6 parameters to determine our local surface model:

$$\mathcal{M} = \{\rho_c(0), \sigma, \nu, \psi, k_1, k_2\}\tag{B.1.29}$$

B.2 Differential Relations: Rigid, Stationary Surface

We now address how the image of a *neighborhood of a point* moves as the camera moves. The neighborhood is first assumed to arise from a smooth surface patch which we locally approximate as a second-order Monge patch around a central point $\mathbf{\Gamma}_c(0) = \mathbf{\Gamma}_{c,0} = \rho_{c,0}\gamma_{c,0} = \rho_{c,0}(\xi_{c,0}, \eta_{c,0}, 1)^\top$. We will also consider the special case of a planar patch.

Theorem B.2.1. (Relating normal and depth gradient) Let γ be the projection of a 3D point $\mathbf{\Gamma}$ lying on a surface with normal \mathbf{N} . Then, when $\mathbf{N} \cdot \gamma \neq 0$, \mathbf{N} and $\frac{\nabla \rho}{\rho}$ are uniquely related as

$$\left\{ \begin{array}{l} \mathbf{N} = \left(-\frac{\nabla \rho}{\rho} (\mathbf{N} \cdot \gamma), \sqrt{1 - (\mathbf{N} \cdot \gamma)^2 \frac{\|\nabla \rho\|^2}{\rho^2}} \right) \end{array} \right. \quad (\text{B.2.1})$$

$$\left\{ \begin{array}{l} \frac{\nabla \rho}{\rho} = -\frac{(\mathbf{N}_x, \mathbf{N}_y)}{\mathbf{N} \cdot \gamma}, \end{array} \right. \quad (\text{B.2.2})$$

Proof. Equation B.2.1 is derived from:

$$\mathbf{N} \cdot \mathbf{N} = 1 \quad (\text{B.2.3})$$

$$(\mathbf{N}_x, \mathbf{N}_y) \cdot (\mathbf{N}_x, \mathbf{N}_y) + \mathbf{N}_z^2 = 1. \quad (\text{B.2.4})$$

Using (B.2.2),

$$\frac{\nabla \rho}{\rho} = -\frac{(\mathbf{N}_x, \mathbf{N}_y)}{\mathbf{N} \cdot \gamma}, \quad (\text{B.2.5})$$

we have

$$(\mathbf{N} \cdot \gamma)^2 \frac{\|\nabla \rho\|^2}{\rho^2} + \mathbf{N}_z^2 = 1, \quad (\text{B.2.6})$$

so that

$$\mathbf{N}_z^2 = 1 - (\mathbf{N} \cdot \gamma)^2 \frac{\|\nabla \rho\|^2}{\rho^2}. \quad (\text{B.2.7})$$

from which the result follows.

We would like to express $\nabla \rho = (\rho_\xi, \rho_\eta)$ explicitly in terms of the first-order surface geometry, which can be represented by the normal \mathbf{N} . Note that the surface under consideration is not $(\xi, \eta, \rho(\xi, \eta))$, but rather $\mathbf{\Gamma}(\xi, \eta) = (\rho(\xi, \eta)\xi, \rho(\xi, \eta)\eta, \rho(\xi, \eta))$. The expression for \mathbf{N} in terms of $\nabla \rho$ is trivial – just compute the derivatives of $\mathbf{\Gamma} = \rho\gamma$ with respect to ξ, η , take their cross product and normalize. However, in order to have the normal explicit in our Equation (3.6.1), we need the inverse, namely $\nabla \rho$ in terms of \mathbf{N} . Differentiating $\mathbf{\Gamma} = \rho\gamma$ we have

$$\left\{ \begin{array}{l} \mathbf{\Gamma}_\xi = \rho_\xi \gamma + \rho \gamma_\xi = \rho_\xi \gamma + \rho \mathbf{e}_1 \\ \mathbf{\Gamma}_\eta = \rho_\eta \gamma + \rho \gamma_\eta = \rho_\eta \gamma + \rho \mathbf{e}_2. \end{array} \right.$$

We can isolate ρ_ξ and ρ_η by taking the scalar product with \mathbf{N}

$$\left\{ \begin{array}{l} 0 = \rho_\xi \mathbf{N} \cdot \gamma + \rho \mathbf{e}_1 \cdot \mathbf{N} \\ 0 = \rho_\eta \mathbf{N} \cdot \gamma + \rho \mathbf{e}_2 \cdot \mathbf{N}, \end{array} \right. \quad (\text{B.2.8})$$

which gives

$$\left\{ \begin{array}{l} \rho_\xi = -\frac{\rho}{\mathbf{N} \cdot \gamma} \mathbf{N}_x \\ \rho_\eta = -\frac{\rho}{\mathbf{N} \cdot \gamma} \mathbf{N}_y \end{array} \right. \quad (\text{B.2.9})$$

■

The velocity vector field specified in Equation 3.5.14 is for a fixed 3D point and depends on the global¹ camera motion parameters $\Omega(t)$ and $\mathbf{V}(t)$ (six numbers for each time) and, assuming this point lies on a surface, locally on depth $\rho(\xi, \eta)$. Thus, all variations of u and v with respect to the spatial variables ξ and η at a fixed time t arise entirely from variations in depth $\rho(\xi, \eta)$, which itself arises from the local shape of the object, as the next theorem shows. This motivates us to relate the spatial derivatives of $\boldsymbol{\chi} = (u, v)$ to those of ρ , as in the theorems that follow.

Proposition B.2.2. (Depth and surface geometry) *Given a paraboloid $\frac{1}{2}\bar{\boldsymbol{\Gamma}}^\top \mathbf{K}\bar{\boldsymbol{\Gamma}} + e_3^\top \bar{\boldsymbol{\Gamma}} = 0$ in the local coordinate system defined by $\boldsymbol{\Gamma}_0 = \bar{R}_0 \bar{\boldsymbol{\Gamma}}_0 + \bar{\mathcal{T}}$, and a camera motion defined by $R(t)$ and $\mathcal{T}(t)$, then the depth $\rho(\xi, \eta)$ satisfies the equation*

$$(\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma}) \rho^2 + \left(e_3^\top \bar{R}^\top \boldsymbol{\gamma} - 2\bar{\mathcal{T}}^\top \mathbf{K}_c \boldsymbol{\gamma} \right) \rho + \left(\bar{\mathcal{T}}^\top \mathbf{K}_c \bar{\mathcal{T}} - e_3^\top \bar{R}^\top \bar{\mathcal{T}} \right) = 0, \quad (\text{B.2.10})$$

where $\mathbf{K}_c = \frac{1}{2} \bar{R} \mathbf{K} \bar{R}^\top$, \mathbf{K} is the principal curvature matrix of the surface as defined in Equation (3.6.8), $\bar{R} = R(t) \bar{R}_0$, and $\bar{\mathcal{T}} = R(t) \bar{\mathcal{T}}_0 + \mathcal{T}(t)$.

Proof. Using $\boldsymbol{\Gamma} = \rho \boldsymbol{\gamma}$ in Equation B.1.17, we have the result. ■

Theorem B.2.3. (Spatial variation of the image velocity field in terms of surface). *Consider an image point $\boldsymbol{\gamma}$ arising from $\boldsymbol{\Gamma}$ lying on a twice-differentiable surface with normal \mathbf{N} and having Weingarten map \mathbf{K}_c in camera coordinates, under differential camera motion Ω, \mathbf{V} . Then the velocity $\boldsymbol{\gamma}_t = (u, v, 1)^\top$ has spatial variation given by Equation 3.6.1:*

$$\begin{cases} \gamma_{\xi t} = (-\mathbf{V} + \mathbf{V}_z \boldsymbol{\gamma}) \frac{\rho_\xi}{\rho^2} - \frac{\mathbf{V}_z}{\rho} \mathbf{e}_1 + \Omega_x \mathbf{e}_1 + \Omega_y \boldsymbol{\gamma} - (e_3^\top \Omega \times \boldsymbol{\gamma}) \mathbf{e}_1 \\ \gamma_{\eta t} = (-\mathbf{V} + \mathbf{V}_z \boldsymbol{\gamma}) \frac{\rho_\eta}{\rho^2} - \frac{\mathbf{V}_z}{\rho} \mathbf{e}_2 + \Omega_x \mathbf{e}_2 + \Omega_x \boldsymbol{\gamma} - (e_3^\top \Omega \times \boldsymbol{\gamma}) \mathbf{e}_2. \end{cases} \quad (\text{B.2.11})$$

Similarly, the second-order spatial derivatives of the image flow (u, v) are given by (3.6.2):

$$\begin{cases} u_{\xi\xi} = (-\mathbf{V}_x + \mathbf{V}_z \xi) \frac{\rho_{\xi\xi}}{\rho^2} - 2(-\mathbf{V}_x + \mathbf{V}_z \xi) \frac{\rho_\xi^2}{\rho^3} + \mathbf{V}_z \frac{\rho_\xi}{\rho^2} + 2\Omega_y \\ u_{\xi\eta} = (-\mathbf{V}_x + \mathbf{V}_z \xi) \frac{\rho_{\xi\eta}}{\rho^2} - 2(-\mathbf{V}_x + \mathbf{V}_z \xi) \frac{\rho_\xi \rho_\eta}{\rho^3} + \mathbf{V}_z \frac{\rho_\eta}{\rho^2} + \Omega_x \\ u_{\eta\eta} = (-\mathbf{V}_x + \mathbf{V}_z \xi) \frac{\rho_{\eta\eta}}{\rho^2} - 2(-\mathbf{V}_x + \mathbf{V}_z \xi) \frac{\rho_\eta^2}{\rho^3} \\ v_{\xi\xi} = (-\mathbf{V}_y + \mathbf{V}_z \eta) \frac{\rho_{\xi\xi}}{\rho^2} - 2(-\mathbf{V}_y + \mathbf{V}_z \eta) \frac{\rho_\xi^2}{\rho^3} \\ v_{\xi\eta} = (-\mathbf{V}_y + \mathbf{V}_z \eta) \frac{\rho_{\xi\eta}}{\rho^2} - 2(-\mathbf{V}_y + \mathbf{V}_z \eta) \frac{\rho_\xi \rho_\eta}{\rho^3} + \mathbf{V}_z \frac{\rho_\xi}{\rho^2} + \Omega_y \\ v_{\eta\eta} = (-\mathbf{V}_y + \mathbf{V}_z \eta) \frac{\rho_{\eta\eta}}{\rho^2} - 2(-\mathbf{V}_y + \mathbf{V}_z \eta) \frac{\rho_\eta^2}{\rho^3} + \mathbf{V}_z \frac{\rho_\eta}{\rho^2} + 2\Omega_x \end{cases} \quad (\text{B.2.12})$$

¹Global in the sense that the parameters are shared by all pixels. Thus their estimation at different points does not introduce new unknowns.

where the normalized gradient of depth is given by:

$$\frac{\nabla \rho}{\rho} = -\frac{(\mathbf{N}_x, \mathbf{N}_y)}{\mathbf{N} \cdot \boldsymbol{\gamma}}, \quad (\text{B.2.13})$$

(observe that for an occluding contour, $\mathbf{N} \cdot \boldsymbol{\gamma} = 0$, the gradient of depth blows up) and the Hessian of depth is given by

$$\begin{cases} \rho_{\xi\xi} = \frac{2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho^2 \mathbf{N}_x^2}{(\mathbf{N} \cdot \boldsymbol{\gamma})^3} + \frac{[4\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_1 + 2\mathbf{N}_x] \rho \mathbf{N}_x}{(\mathbf{N} \cdot \boldsymbol{\gamma})^2} + \frac{2e_1^\top \mathbf{K}_c e_1 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \\ \rho_{\eta\eta} = \frac{2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho^2 \mathbf{N}_y^2}{(\mathbf{N} \cdot \boldsymbol{\gamma})^3} + \frac{[4\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_2 + 2\mathbf{N}_y] \rho \mathbf{N}_y}{(\mathbf{N} \cdot \boldsymbol{\gamma})^2} + \frac{2e_2^\top \mathbf{K}_c e_2 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \\ \rho_{\xi\eta} = \frac{2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho^2 \mathbf{N}_x \mathbf{N}_y}{(\mathbf{N} \cdot \boldsymbol{\gamma})^3} + \frac{[2\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_2 + \mathbf{N}_y] \rho \mathbf{N}_x + [2\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_1 + \mathbf{N}_x] \rho \mathbf{N}_y}{(\mathbf{N} \cdot \boldsymbol{\gamma})^2} + \frac{2e_2^\top \mathbf{K}_c e_1 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \end{cases} \quad (\text{B.2.14})$$

where $\mathbf{K}_c = \frac{1}{2} \bar{\mathbf{R}} \mathbf{K} \bar{\mathbf{R}}^\top$, \mathbf{K} is the principal curvature matrix of the surface as defined in Equation (3.6.8), $\bar{\mathbf{R}} = R(t) \bar{\mathbf{R}}_0$, and $\bar{\mathbf{T}} = R(t) \bar{\mathbf{T}}_0 + \mathcal{T}(t)$.

Proof. For the first-order image motion field, Use the general Equation 3.5.54 for an image curve given in Theorem 3.5.11, where here the image curve is taken to be along one of the coordinate axes, so that the parameter s can be taken as ξ or η . The formula admits the specific form given in this theorem, by observing that $\boldsymbol{\gamma}_\xi = \mathbf{e}_1$ and $\boldsymbol{\gamma}_\eta = \mathbf{e}_2$.

The second spatial derivatives of ρ are computed by considering an explicit osculating paraboloid equation as in the next section. We can then differentiate Equation (B.2.10) with respect to ξ, η and evaluate the result at the center point, to get:

$$\rho_{\xi\xi} = \frac{[4\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_1 + 2\mathbf{N}_x] \rho_\xi + 2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho_\xi^2 + 2e_1^\top \mathbf{K}_c e_1 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \quad (\text{B.2.15})$$

$$\rho_{\eta\eta} = \frac{[4\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_2 + 2\mathbf{N}_y] \rho_\eta + 2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho_\eta^2 + 2e_2^\top \mathbf{K}_c e_2 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}} \quad (\text{B.2.16})$$

$$\rho_{\xi\eta} = \frac{[2\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_2 + \mathbf{N}_y] \rho_\xi + [2\rho\boldsymbol{\gamma}^\top \mathbf{K}_c e_1 + \mathbf{N}_x] \rho_\eta + 2\boldsymbol{\gamma}^\top \mathbf{K}_c \boldsymbol{\gamma} \rho_\eta \rho_\xi + 2e_2^\top \mathbf{K}_c e_1 \rho^2}{\mathbf{N} \cdot \boldsymbol{\gamma}}, \quad (\text{B.2.17})$$

where $\mathbf{K}_c = \frac{1}{2} \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top$ and ρ_ξ, ρ_η are given by (B.2.2). Substituting the above values into (3.6.2), we get the desired second-order derivatives of flow. ■

Proposition B.2.4. *The depth ρ is related to slant angle σ by*

$$\tan \sigma = \frac{\|\nabla \rho\|}{\rho}. \quad (\text{B.2.18})$$

The second-order variation of flow is computed in a similar way. Thus, the second-order variation of the velocity vector field is a function of the six global motion parameters (Ω, \mathbf{V}) , the depth ρ at (ξ, η) , the depth gradient $\nabla \rho = (\rho_\xi, \rho_\eta)$ (which are derived from the normal to the surface), and finally, the depth Hessian

$$\text{Hessian} = \begin{bmatrix} \rho_{\xi\xi} & \rho_{\xi\eta} \\ \rho_{\eta\xi} & \rho_{\eta\eta} \end{bmatrix}. \quad (\text{B.2.19})$$

We related the Hessian to the principal curvatures κ_1, κ_2 and the angle of the larger principal curvature with the x -axis. Thus, six shape variables, the depth ρ , the depth surface normal (2) and the surface curvature (3), together with the six motion parameters determine up to the second-order variation of the velocity vector field.

Remark B.2.1. We can write Equation B.2.10 more explicitly as:

$$q_1x^2 + q_2y^2 + q_3z^2 + q_4xy + q_5xz + q_6yz + q_7x + q_8y + q_9z + q_{10} = 0, \quad (\text{B.2.20})$$

where the coefficients q_i for $t = 0$ are given in Section B.5. Substituting equation (B.1.21), we have:

$$q_1\xi^2\rho^2 + q_2\eta^2\rho^2 + q_3\rho^2 + q_4\xi\eta\rho^2 + q_5\xi\rho^2 + q_6\eta\rho^2 + q_7\xi\rho + q_8\eta\rho + q_9\rho + q_{10} = 0, \quad (\text{B.2.21})$$

or, by grouping ρ :

$$(q_1\xi^2 + q_2\eta^2 + q_4\xi\eta + q_5\xi + q_6\eta + q_3)\rho^2 + (q_7\xi + q_8\eta + q_9)\rho + q_{10} = 0. \quad (\text{B.2.22})$$

If we had $q_{10} = 0$ (a similar case as Shashua's paper), we could trivially solve for $\rho(\xi, \eta)$. However, in our case, we prefer not to solve this explicitly, as it would lead to square roots. Rather, we keep this equation as our model constraint on $\rho(\xi, \eta)$.

Remark B.2.2. Plugging ρ_0 from (3.5.15), into the model equation (B.2.22), we will have one polynomial equation in ξ, η, u, v , at $t = 0$. It is quadratic in u , giving two possible solutions. For each of these, the epipolar constraint gives one v , so we have two possible solutions for u and v . Instead of expressing these solutions explicitly, we keep the epipolar equation and the model equation to be solved jointly with correspondence constraints such as optical flow.

Corollary B.2.5. (planar flow) *The image motion field induced by a plane is given by:*

$$\begin{cases} u = (h_1\xi^2 + h_2\xi\eta + h_3\xi + h_4\eta + h_5)/h_6 \\ v = (h_2\eta^2 + h_1\xi\eta + h_7\xi + h_8\eta + h_9)/h_6 \end{cases} \quad (\text{B.2.23})$$

where the coefficients h_i are listed in Section B.5.

Proof. Setting $\kappa_1 = \kappa_2 = 0$ in Equation (B.2.10), we get

$$\left(e_3^\top \overline{R}^\top \gamma\right) \rho - \left(e_3^\top \overline{R}^\top \overline{T}\right) = 0, \quad (\text{B.2.24})$$

and, isolating ρ ,

$$\rho = \frac{e_3^\top \overline{R}^\top \overline{T}}{e_3^\top \overline{R}^\top \gamma}. \quad (\text{B.2.25})$$

Plugging into the flow equations (3.5.14), we have the desired result. ■

B.3 Shape from Spatio-Temporal Shading

Let us assume that an image patch Ω around (ξ_c, η_c) corresponds to surface \mathcal{M} in space locally approximated by a paraboloid, as in the previous section. The surface is assumed to be Lambertian with BRDF α . Moreover, we assume a point-source of light having intensity l whose position is $\mathbf{L} = [L_1, L_2, L_3]^\top$ in coordinates of the camera at $t = 0$. The irradiance equation can be written as:

$$I(\xi, \eta) = \alpha l \frac{(\mathbf{L} - \mathbf{\Gamma})^\top}{\|\mathbf{L} - \mathbf{\Gamma}\|} \overline{R_0 \mathbf{N}} \quad (\text{B.3.1})$$

where $\overline{\mathbf{N}} = [\overline{N_x}, \overline{N_y}, \overline{N_z}]^\top$ is the surface normal in local coordinates at the point in \mathcal{M} corresponding to (ξ, η) , and $\mathbf{\Gamma}$ is the coordinates of such point in the camera frame at $t = 0$, and we can write $\mathbf{\Gamma} = \rho \boldsymbol{\gamma}$. We can use the simplification that, for all ξ, η near ξ_c, η_c , the light direction $\mathbf{L} - \mathbf{\Gamma}$ is approximately $\mathbf{L} - \mathbf{\Gamma}_c$.

The photometric parameters α, \mathbf{L} can be assumed constant within the patch Ω , or even globally. Of course, α will change for reflectance edges and across different surfaces, while \mathbf{L} will change in case there are multiple light sources.

We can also define the above equation for colored light. An approximation is to use three primary colors RGB. We would then have three light intensities l_r, l_g, l_b , and three $\alpha_r, \alpha_g, \alpha_b$. Our equation could then be written as:

$$\begin{bmatrix} I_r(\xi, \eta) \\ I_g(\xi, \eta) \\ I_b(\xi, \eta) \end{bmatrix} = \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{bmatrix} \begin{bmatrix} l_r \\ l_g \\ l_b \end{bmatrix} \frac{(\mathbf{L} - \mathbf{\Gamma})^\top}{\|\mathbf{L} - \mathbf{\Gamma}\|} \overline{R_0 \mathbf{N}} \quad (\text{B.3.2})$$

$$\mathbf{I}(\xi, \eta) = \boldsymbol{\alpha} l \frac{(\mathbf{L} - \mathbf{\Gamma})^\top}{\|\mathbf{L} - \mathbf{\Gamma}\|} \overline{R_0 \mathbf{N}} \quad (\text{B.3.3})$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{bmatrix}, \quad \mathbf{l} = \begin{bmatrix} l_r \\ l_g \\ l_b \end{bmatrix}. \quad (\text{B.3.4})$$

We are concerned with the following problem. Given the image RGB intensity vector \mathbf{I} , and its derivatives $\mathbf{I}_\xi, \mathbf{I}_\eta, \mathbf{I}_{\xi\xi}, \mathbf{I}_{\xi\eta}, \mathbf{I}_{\eta\eta}, \mathbf{I}_t, \mathbf{I}_{\xi t}, \mathbf{I}_{\eta t}, \mathbf{I}_{tt}$ measured at the image point (ξ_c, η_c) at time t , the goal is to recover 20 parameters: 6 local surface parameters \mathcal{M} within the patch Ω centered on (ξ_c, η_c) , 6 global infinitesimal camera motion parameters $\Omega_\times, \mathcal{T}(t)$ and 8 global photometric parameters $\boldsymbol{\rho}, \mathbf{l}$, and \mathbf{L} .

In order to solve such problem, it is necessary to compute the derivatives of normal, resulting in

the following 10 equations:

$$\mathbf{I}(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}} \quad (\text{B.3.5})$$

$$\mathbf{I}_\xi(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_\xi \quad (\text{B.3.6})$$

$$\mathbf{I}_\eta(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_\eta \quad (\text{B.3.7})$$

$$\mathbf{I}_{\xi\xi}(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_{\xi\xi} \quad (\text{B.3.8})$$

$$\mathbf{I}_{\xi\eta}(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_{\xi\eta} \quad (\text{B.3.9})$$

$$\mathbf{I}_{\eta\eta}(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_{\eta\eta} \quad (\text{B.3.10})$$

$$\mathbf{I}_t(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_t \quad (\text{B.3.11})$$

$$\mathbf{I}_{tt}(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_{tt} \quad (\text{B.3.12})$$

$$\mathbf{I}_{\xi t}(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_{\xi t} \quad (\text{B.3.13})$$

$$\mathbf{I}_{\eta t}(\xi, \eta) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{R}_0 \bar{\mathbf{N}}_{\eta t} \quad (\text{B.3.14})$$

where we are actually interested in the case where $t = 0$, $\xi = \xi_c$, $\eta = \eta_c$, and, thus, $\bar{x} = 0$ and $\bar{y} = 0$. Note we are also assuming that the light direction is constant for the whole patch, meaning $\mathbf{L} - \boldsymbol{\Gamma}$ is approximated by $\mathbf{L} - \boldsymbol{\Gamma}_c(0) = \mathbf{L} - \rho_c(0)\boldsymbol{\gamma}_c(0)$, and that the derivatives of $(\mathbf{L} - \boldsymbol{\Gamma}_c)$ are 0.

The normal can be written in local coordinates as:

$$\bar{\mathbf{N}}(\bar{x}, \bar{y}) = \frac{[-k_1 \bar{x}, -k_2 \bar{y}, 1]^\top}{\sqrt{k_1^2 \bar{x}^2 + k_2^2 \bar{y}^2 + 1}}, \quad (\text{B.3.15})$$

or, setting $\bar{x} = 0$, and $\bar{y} = 0$

$$\bar{\mathbf{N}}(0, 0) = [0, 0, 1]^\top = \mathbf{e}_3. \quad (\text{B.3.16})$$

The following is a summary of the above formulas, in terms of the unknowns of the problem. Their

derivation is listed in Section B.4. At $t = 0$ we have:

$$\mathbf{I}(\xi_c, \eta_c) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \mathbf{e}_3 \quad (\text{B.3.17})$$

$$\mathbf{I}_\xi(\xi_c, \eta_c) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_\xi \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_1) \quad (\text{B.3.18})$$

$$\mathbf{I}_\eta(\xi_c, \eta_c) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_\eta \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_2) \quad (\text{B.3.19})$$

$$\mathbf{I}_{\xi\xi}(\xi_c, \eta_c) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \left\{ \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_{\xi\xi} \boldsymbol{\gamma}_c + 2\rho_\xi \mathbf{e}_1) - [(\rho_\xi \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_1)^\top \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_\xi \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_1)] \mathbf{e}_3 \right\} \quad (\text{B.3.20})$$

$$\mathbf{I}_{\eta\eta}(\xi_c, \eta_c) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \left\{ \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_{\eta\eta} \boldsymbol{\gamma}_c + 2\rho_\eta \mathbf{e}_2) - [(\rho_\eta \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_2)^\top \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_\eta \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_2)] \mathbf{e}_3 \right\} \quad (\text{B.3.21})$$

$$\mathbf{I}_{\xi\eta}(\xi_c, \eta_c) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \left\{ \begin{aligned} & \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_{\xi\eta} \boldsymbol{\gamma}_c + \rho_\xi \mathbf{e}_2 + \rho_\eta \mathbf{e}_1) + \\ & - [(\rho_\eta \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_2)^\top \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_\xi \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_1)] \mathbf{e}_3 \end{aligned} \right\} \quad (\text{B.3.22})$$

$$\mathbf{I}_t(\xi_c, \eta_c) = \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top [\rho_t \boldsymbol{\gamma}_c - \Omega_\times \rho_c \boldsymbol{\gamma}_c - \mathcal{T}_t] \quad (\text{B.3.23})$$

$$\begin{aligned} \mathbf{I}_{tt}(\xi_c, \eta_c) &= \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \{ \mathbf{K} \bar{\mathbf{R}}_0^\top [\rho_{tt} \boldsymbol{\gamma}_c - 2\Omega_\times \rho_t \boldsymbol{\gamma}_c + 2\Omega_\times \Omega_\times \rho_c \boldsymbol{\gamma}_c + 2\Omega_\times \mathcal{T}_t] \\ &\quad - [(\rho_t \boldsymbol{\gamma}_c - \Omega_\times \rho_c \boldsymbol{\gamma}_c - \mathcal{T}_t)^\top \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_t \boldsymbol{\gamma}_c - \Omega_\times \rho_c \boldsymbol{\gamma}_c - \mathcal{T}_t)] \mathbf{e}_3 \} \end{aligned} \quad (\text{B.3.24})$$

$$\begin{aligned} \mathbf{I}_{\xi t}(\xi_c, \eta_c) &= \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \{ \mathbf{K} \bar{\mathbf{R}}_0^\top [(\rho_{\xi t} \boldsymbol{\gamma}_c + \rho_t \mathbf{e}_1) - \Omega_\times (\rho_\xi \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_1)] \\ &\quad - [(\rho_t \boldsymbol{\gamma}_c - \Omega_\times \rho_c \boldsymbol{\gamma}_c - \mathcal{T}_t)^\top \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_\xi \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_1)] \mathbf{e}_3 \} \end{aligned} \quad (\text{B.3.25})$$

$$\begin{aligned} \mathbf{I}_{\eta t}(\xi_c, \eta_c) &= \boldsymbol{\rho} \mathbf{l} \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \{ \mathbf{K} \bar{\mathbf{R}}_0^\top [(\rho_{\eta t} \boldsymbol{\gamma}_c + \rho_t \mathbf{e}_2) - \Omega_\times (\rho_\eta \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_2)] \\ &\quad - [(\rho_t \boldsymbol{\gamma}_c - \Omega_\times \rho_c \boldsymbol{\gamma}_c - \mathcal{T}_t)^\top \bar{\mathbf{R}}_0 \mathbf{K} \bar{\mathbf{R}}_0^\top (\rho_\eta \boldsymbol{\gamma}_c + \rho_c \mathbf{e}_2)] \mathbf{e}_3 \} \end{aligned} \quad (\text{B.3.26})$$

where the derivatives of ρ can be obtained from the model equation (B.1.25). All variables are

evaluated at $t = 0$. Letting $\mathbf{K}_c = \frac{1}{2}\bar{\mathbf{R}}_0\mathbf{K}\bar{\mathbf{R}}_0^\top$, the derivatives of ρ at $t = 0$ are:

$$\rho_\xi = \frac{-\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \mathbf{e}_1 \rho_c}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \quad (\text{B.3.27})$$

$$\rho_\eta = \frac{-\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \mathbf{e}_2 \rho_c}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \quad (\text{B.3.28})$$

$$\rho_{\xi\xi} = \frac{\left[4\rho_c \boldsymbol{\gamma}_c^\top \mathbf{K}_c \mathbf{e}_1 + 2\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \mathbf{e}_1\right] \rho_\xi + 2\boldsymbol{\gamma}_c^\top \mathbf{K}_c \boldsymbol{\gamma}_c \rho_\xi^2 + 2\mathbf{e}_1^\top \mathbf{K}_c \mathbf{e}_1 \rho_c^2}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \quad (\text{B.3.29})$$

$$\rho_{\eta\eta} = \frac{\left[4\rho_c \boldsymbol{\gamma}_c^\top \mathbf{K}_c \mathbf{e}_2 + 2\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \mathbf{e}_2\right] \rho_\eta + 2\boldsymbol{\gamma}_c^\top \mathbf{K}_c \boldsymbol{\gamma}_c \rho_\eta^2 + 2\mathbf{e}_2^\top \mathbf{K}_c \mathbf{e}_2 \rho_c^2}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \quad (\text{B.3.30})$$

$$\rho_{\xi\eta} = \frac{\left[2\rho_c \boldsymbol{\gamma}_c^\top \mathbf{K}_c \mathbf{e}_2 + \mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \mathbf{e}_2\right] \rho_\xi + \left[2\rho_c \boldsymbol{\gamma}_c^\top \mathbf{K}_c \mathbf{e}_1 + \mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \mathbf{e}_1\right] \rho_\eta + 2\boldsymbol{\gamma}_c^\top \mathbf{K}_c \boldsymbol{\gamma}_c \rho_\eta \rho_\xi + 2\mathbf{e}_2^\top \mathbf{K}_c \mathbf{e}_1 \rho_c^2}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \quad (\text{B.3.31})$$

$$\rho_t = \frac{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top (\Omega_\times \rho_c \boldsymbol{\gamma}_c + \mathcal{T}_t)}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \quad (\text{B.3.32})$$

$$\rho_{\xi t} = \frac{-1}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \left(- \left[\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \Omega_\times \boldsymbol{\gamma}_c + \bar{\mathcal{T}}_t^\top 2\mathbf{K}_c \boldsymbol{\gamma}_c \right] \rho_\xi + 2\boldsymbol{\gamma}_c^\top \mathbf{K}_c \boldsymbol{\gamma}_c \rho_t \rho_\xi - \left[\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \Omega_\times \mathbf{e}_1 + \bar{\mathcal{T}}_t^\top 2\mathbf{K}_c \mathbf{e}_1 \right] \rho_c \right) \quad (\text{B.3.33})$$

$$\rho_{\eta t} = \frac{-1}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \left(- \left[\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \Omega_\times \boldsymbol{\gamma}_c + \bar{\mathcal{T}}_t^\top 2\mathbf{K}_c \boldsymbol{\gamma}_c \right] \rho_\eta + 2\boldsymbol{\gamma}_c^\top \mathbf{K}_c \boldsymbol{\gamma}_c \rho_t \rho_\eta - \left[\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \Omega_\times \mathbf{e}_2 + \bar{\mathcal{T}}_t^\top 2\mathbf{K}_c \mathbf{e}_2 \right] \rho_c \right) \quad (\text{B.3.34})$$

$$\rho_{tt} = \frac{1}{\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \boldsymbol{\gamma}_c} \left(-2\boldsymbol{\gamma}_c^\top \mathbf{K}_c \boldsymbol{\gamma}_c \rho_t^2 - 2 \left[\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \Omega_\times \boldsymbol{\gamma}_c - \bar{\mathcal{T}}_t^\top 2\mathbf{K}_c \boldsymbol{\gamma}_c \right] \rho_t + \left[+2\boldsymbol{\gamma}_c^\top \Omega_\times \mathbf{K}_c \Omega_\times \boldsymbol{\gamma}_c \rho_c^2 - 2\bar{\mathcal{T}}_t^\top \mathbf{K}_c \bar{\mathcal{T}}_t - 2\rho_c \boldsymbol{\gamma}_c^\top \Omega_\times \mathbf{K}_c \bar{\mathcal{T}}_t + 2\mathbf{e}_3^\top \bar{\mathbf{R}}_0^\top \Omega_\times \bar{\mathcal{T}}_t \right] \rho_c \right) \quad (\text{B.3.35})$$

where

$$\bar{\mathcal{T}}_t = (\Omega_\times \rho_c \boldsymbol{\gamma}_c + \mathcal{T}_t), \text{ and } \ddot{\bar{\mathcal{T}}} = \ddot{\mathcal{T}} \quad (\text{B.3.36})$$

Eliminating variables αl The term $\frac{\alpha l}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|}$ appears in all equations. We can eliminate it, at least in the monochromatic case, using the first equation to solve for it, ending up with 9 remaining equations. From (B.3.17) and any other equation of the form:

$$I_{xx} = \alpha l \frac{(\mathbf{L} - \rho_c \boldsymbol{\gamma}_c)^\top}{\|\mathbf{L} - \rho_c \boldsymbol{\gamma}_c\|} \bar{\mathbf{R}}_0 \bar{\mathbf{N}}_{xx} \quad (\text{B.3.37})$$

(where we denote by I_{xx} and $\bar{\mathbf{N}}_{xx}$ any of the derivatives of I and $\bar{\mathbf{N}}$, resp.) we can cross-multiply and get:

$$I\alpha l \frac{(\mathbf{L} - \rho_c \gamma_c)^\top}{\|\mathbf{L} - \rho_c \gamma_c\|} \bar{R}_0 \bar{\mathbf{N}}_{xx} = I_{xx} \alpha l \frac{(\mathbf{L} - \rho_c \gamma_c)^\top}{\|\mathbf{L} - \rho_c \gamma_c\|} \bar{R}_0 e_3 \quad (\text{B.3.38})$$

$$I(\mathbf{L} - \rho_c \gamma_c)^\top \bar{R}_0 \bar{\mathbf{N}}_{xx} = I_{xx} (\mathbf{L} - \rho_c \gamma_c)^\top \bar{R}_0 e_3 \quad (\text{B.3.39})$$

thus

$$(\mathbf{L} - \rho_c \gamma_c)^\top \bar{R}_0 [I_{xx} e_3 - I \bar{\mathbf{N}}_{xx}] = 0 \quad (\text{B.3.40})$$

$$\mathbf{L}^\top \bar{R}_0 [I \bar{\mathbf{N}}_{xx} - I_{xx} e_3] - \rho_c \gamma_c^\top \bar{R}_0 [I \bar{\mathbf{N}}_{xx} - I_{xx} e_3] = 0 \quad (\text{B.3.41})$$

Now, 3 such equations yield \mathbf{L} in terms of all other unknowns. Since \mathbf{L} is global, we can pick such equations from different patches, so all other patches will still have all 9 equations to work with.

B.3.1 Considering Intrinsic Parameters

In practice, we cannot measure normalized image coordinates directly $\gamma = [\xi, \eta, 1]^\top$, but only in terms of image pixel coordinates $\gamma_{im} = [x_{im}, y_{im}, 1]^\top$. Let the intrinsic parameter matrix be K_{im} :

$$K_{im} = \begin{bmatrix} \alpha_\xi & s & \xi_o \\ 0 & \alpha_\eta & \eta_o \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.3.42})$$

Then we have:

$$\gamma_{im} = K_{im} \gamma \quad (\text{B.3.43})$$

$$\gamma = K_{im}^{-1} \gamma_{im} \quad (\text{B.3.44})$$

That is how one transforms from image pixel coordinates to normalized image coordinates. More explicitly,

$$\begin{cases} \xi = \frac{1}{\alpha_\xi} x_{im} - \frac{s}{\alpha_\xi \alpha_\eta} y_{im} + \frac{s \eta_o}{\alpha_\xi \alpha_\eta} - \frac{\xi_o}{\alpha_\xi} \\ \eta = \frac{1}{\alpha_\eta} y_{im} - \frac{\eta_o}{\alpha_\eta}, \end{cases} \quad (\text{B.3.45})$$

$$\quad (\text{B.3.46})$$

with

$$K_{im}^{-1} = \begin{bmatrix} \frac{1}{\alpha_\xi} & -\frac{s}{\alpha_\xi \alpha_\eta} & \frac{s \eta_o}{\alpha_\xi \alpha_\eta} - \frac{\xi_o}{\alpha_\xi} \\ 0 & \frac{1}{\alpha_\eta} & -\frac{\eta_o}{\alpha_\eta} \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.3.47})$$

The derivatives are:

$$\mathbf{I}_\xi = \alpha_\xi \mathbf{I}_{x_{im}} \quad (\text{B.3.48})$$

$$\mathbf{I}_\eta = s \mathbf{I}_{x_{im}} + \alpha_\eta \mathbf{I}_{y_{im}} \quad (\text{B.3.49})$$

$$\mathbf{I}_{\xi\xi} = \alpha_\xi^2 \mathbf{I}_{x_{im}x_{im}} \quad (\text{B.3.50})$$

$$\mathbf{I}_{\xi\eta} = \alpha_\xi s \mathbf{I}_{x_{im}x_{im}} + \alpha_\xi \alpha_\eta \mathbf{I}_{x_{im}y_{im}} \quad (\text{B.3.51})$$

$$\mathbf{I}_{\eta\eta} = s^2 \mathbf{I}_{x_{im}x_{im}} + 2s\alpha_\eta \mathbf{I}_{x_{im}y_{im}} + \alpha_\eta^2 \mathbf{I}_{y_{im}y_{im}} \quad (\text{B.3.52})$$

So one must first go through these formulas before using main ones which require normalized image coordinates.

B.4 Derivation of the Shading Differential Equations

In this supplementary section we derive the shading equations listed in Section B.3.

Theorem B.4.1. *The derivatives of normal corresponding to the image point γ_c at $t = 0$ are:*

$$\bar{\mathbf{N}}_\xi = \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \quad (\text{B.4.1})$$

$$\bar{\mathbf{N}}_\eta = \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \eta} \quad (\text{B.4.2})$$

$$\bar{\mathbf{N}}_{\xi\xi} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial \xi^2} - \left[\frac{\partial \bar{\Gamma}}{\partial \xi}^\top \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \right] e_3 \quad (\text{B.4.3})$$

$$\bar{\mathbf{N}}_{\eta\eta} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial \eta^2} - \left[\frac{\partial \bar{\Gamma}}{\partial \eta}^\top \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \eta} \right] e_3 \quad (\text{B.4.4})$$

$$\bar{\mathbf{N}}_{\xi\eta} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial \eta \partial \xi} - \left[\frac{\partial \bar{\Gamma}}{\partial \eta}^\top \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \right] e_3 \quad (\text{B.4.5})$$

$$\bar{\mathbf{N}}_t = \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial t} \quad (\text{B.4.6})$$

$$\bar{\mathbf{N}}_{tt} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial t^2} - \left[\frac{\partial \bar{\Gamma}}{\partial t}^\top \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial t} \right] e_3 \quad (\text{B.4.7})$$

$$\bar{\mathbf{N}}_{\xi t} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial t \partial \xi} - \left[\frac{\partial \bar{\Gamma}}{\partial t}^\top \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \right] e_3 \quad (\text{B.4.8})$$

$$\bar{\mathbf{N}}_{\eta t} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial t \partial \eta} - \left[\frac{\partial \bar{\Gamma}}{\partial t}^\top \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \eta} \right] e_3 \quad (\text{B.4.9})$$

Where

$$\mathbf{K} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{B.4.10})$$

$$\bar{\Gamma}_\xi = \bar{R}_0^\top (\rho_\xi \gamma_c + \rho_c e_1) \quad (\text{B.4.11})$$

$$\bar{\Gamma}_\eta = \bar{R}_0^\top (\rho_\eta \gamma_c + \rho_c e_2) \quad (\text{B.4.12})$$

$$\bar{\Gamma}_{\xi\xi} = \bar{R}_0^\top (\rho_{\xi\xi} \gamma_c + 2\rho_\xi e_1) \quad (\text{B.4.13})$$

$$\bar{\Gamma}_{\eta\eta} = \bar{R}_0^\top (\rho_{\eta\eta} \gamma_c + 2\rho_\eta e_2) \quad (\text{B.4.14})$$

$$\bar{\Gamma}_{\xi\eta} = \bar{R}_0^\top (\rho_{\xi\eta} \gamma_c + \rho_\xi e_2 + \rho_\eta e_1) \quad (\text{B.4.15})$$

$$\bar{\Gamma}_t = \bar{R}_0^\top [\rho_t \gamma_c - \Omega_\times \rho_c \gamma_c - \dot{T}] \quad (\text{B.4.16})$$

$$\bar{\Gamma}_{\xi t} = \bar{R}_0^\top [(\rho_{\xi t} \gamma_c + \rho_t e_1) - \Omega_\times (\rho_\xi \gamma_c + \rho_c e_1)] \quad (\text{B.4.17})$$

$$\bar{\Gamma}_{\eta t} = \bar{R}_0^\top [(\rho_{\eta t} \gamma_c + \rho_t e_2) - \Omega_\times (\rho_\eta \gamma_c + \rho_c e_2)] \quad (\text{B.4.18})$$

$$\bar{\Gamma}_{tt} = \bar{R}_0^\top [\rho_{tt} \gamma_c - 2\Omega_\times \rho_t \gamma_c + 2\Omega_\times \Omega_\times \rho_c \gamma_c + 2\Omega_\times \dot{T}] \quad (\text{B.4.19})$$

Computing $\bar{\mathbf{N}}_\xi$ and $\bar{\mathbf{N}}_\eta$.

$$\bar{\mathbf{N}}_\xi = \bar{\mathbf{N}}_{\bar{x}} \frac{\partial \bar{x}}{\partial \xi} + \bar{\mathbf{N}}_{\bar{y}} \frac{\partial \bar{y}}{\partial \xi} \quad (\text{B.4.20})$$

Now, the Rodrigues formula from differential geometry tells us that $d\bar{\mathbf{N}}$ along a principal direction \bar{x} or \bar{y} is equal to $-kd\bar{\Gamma}$, so we can write:

$$\bar{\mathbf{N}}_\xi = -k_1 \frac{\partial \bar{\Gamma}}{\partial \bar{x}} \frac{\partial \bar{x}}{\partial \xi} - k_2 \frac{\partial \bar{\Gamma}}{\partial \bar{y}} \frac{\partial \bar{y}}{\partial \xi} \quad (\text{B.4.21})$$

$$= -k_1 \begin{bmatrix} 1 \\ 0 \\ k_1 \bar{x} \end{bmatrix} \frac{\partial \bar{x}}{\partial \xi} - k_2 \begin{bmatrix} 0 \\ 1 \\ k_2 \bar{y} \end{bmatrix} \frac{\partial \bar{y}}{\partial \xi} \quad (\text{B.4.22})$$

Setting $\bar{x} = 0$ and $\bar{y} = 0$, we have:

$$\bar{\mathbf{N}}_\xi = \begin{bmatrix} -k_1 \\ 0 \\ 0 \end{bmatrix} \frac{\partial \bar{x}}{\partial \xi} + \begin{bmatrix} 0 \\ -k_2 \\ 0 \end{bmatrix} \frac{\partial \bar{y}}{\partial \xi} = \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \quad (\text{B.4.23})$$

where

$$\mathbf{K} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -k_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{B.4.24})$$

Which can be rewritten by the following consideration:

$$\bar{\Gamma} = \bar{R}^\top \Gamma - \bar{R}^\top \bar{T} \quad (\text{B.4.25})$$

$$\frac{\partial \bar{\Gamma}}{\partial \xi} = \bar{R}^\top \frac{\partial \Gamma}{\partial \xi} \quad (\text{B.4.26})$$

$$\Gamma = \rho \gamma \quad (\text{B.4.27})$$

$$\Gamma_\xi = \rho_\xi \gamma + \rho \gamma_\xi \quad (\text{B.4.28})$$

$$= \rho_\xi \gamma + \rho e_1 \quad (\text{B.4.29})$$

Thus, the final formula for $\bar{x} = \bar{y} = 0, t = 0$ is:

$$\bar{\mathbf{N}}_\xi = \mathbf{K}\bar{R}_0^\top \left(\frac{\partial \rho}{\partial \xi} \gamma_c + \rho_c e_1 \right) \quad (\text{B.4.30})$$

and, similarly

$$\bar{\mathbf{N}}_\eta = \mathbf{K}\bar{R}_0^\top \left(\frac{\partial \rho}{\partial \eta} \gamma_c + \rho_c e_2 \right) \quad (\text{B.4.31})$$

In order to compute derivatives of ρ , we use the model equation (B.1.25). In the end of this section, we list such derivatives necessary for the normal derivative formulas.

Computing $\bar{\mathbf{N}}_{\xi\xi}$ and $\bar{\mathbf{N}}_{\eta\eta}$. From the previous calculations, we know

$$\bar{\mathbf{N}}_\xi = -k_1 \frac{\partial \bar{\Gamma}}{\partial \bar{x}} \frac{\partial \bar{x}}{\partial \xi} - k_2 \frac{\partial \bar{\Gamma}}{\partial \bar{y}} \frac{\partial \bar{y}}{\partial \xi} \quad (\text{B.4.32})$$

thus,

$$\bar{\mathbf{N}}_{\xi\xi} = -k_1 \frac{\partial}{\partial \xi} \left(\frac{\partial \bar{\Gamma}}{\partial \bar{x}} \right) \frac{\partial \bar{x}}{\partial \xi} - k_1 \frac{\partial \bar{\Gamma}}{\partial \bar{x}} \frac{\partial^2 \bar{x}}{\partial \xi^2} \quad (\text{B.4.33})$$

$$\begin{aligned} & - k_2 \frac{\partial}{\partial \xi} \left(\frac{\partial \bar{\Gamma}}{\partial \bar{y}} \right) \frac{\partial \bar{y}}{\partial \xi} - k_2 \frac{\partial \bar{\Gamma}}{\partial \bar{y}} \frac{\partial^2 \bar{y}}{\partial \xi^2} \\ & = -k_1 \frac{\partial^2 \bar{\Gamma}}{\partial \bar{x}^2} \left(\frac{\partial \bar{x}}{\partial \xi} \right)^2 - k_2 \frac{\partial^2 \bar{\Gamma}}{\partial \bar{y}^2} \left(\frac{\partial \bar{y}}{\partial \xi} \right)^2 \end{aligned} \quad (\text{B.4.34})$$

$$\begin{aligned} & - k_1 \frac{\partial \bar{\Gamma}}{\partial \bar{x}} \frac{\partial^2 \bar{x}}{\partial \xi^2} - k_2 \frac{\partial \bar{\Gamma}}{\partial \bar{y}} \frac{\partial^2 \bar{y}}{\partial \xi^2} \\ & = -k_1 \begin{bmatrix} 0 \\ 0 \\ k_1 \end{bmatrix} \left(\frac{\partial \bar{x}}{\partial \xi} \right)^2 - k_2 \begin{bmatrix} 0 \\ 0 \\ k_2 \end{bmatrix} \left(\frac{\partial \bar{y}}{\partial \xi} \right)^2 \end{aligned} \quad (\text{B.4.35})$$

$$- k_1 \begin{bmatrix} 1 \\ 0 \\ k_1 \bar{x} \end{bmatrix} \frac{\partial^2 \bar{x}}{\partial \xi^2} - k_2 \begin{bmatrix} 0 \\ 1 \\ k_2 \bar{y} \end{bmatrix} \frac{\partial^2 \bar{y}}{\partial \xi^2}$$

Hence, at $\bar{x} = \bar{y} = t = 0$, we can write the final formula:

$$\bar{\mathbf{N}}_{\xi\xi} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial \xi^2} - \left[\frac{\partial \bar{\Gamma}}{\partial \xi} \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \right] e_3 \quad (\text{B.4.36})$$

where

$$\frac{\partial^2 \bar{\Gamma}}{\partial \xi^2} = \bar{R}_0^\top [\rho_{\xi\xi} \gamma_c + 2\rho_\xi e_1] \quad (\text{B.4.37})$$

$$\frac{\partial \bar{\Gamma}}{\partial \xi} = \bar{R}_0^\top [\rho_\xi \gamma_c + \rho_c e_1] \quad (\text{B.4.38})$$

and $\rho_{\xi\xi}$ can be obtained from the model equation (B.1.25). Using straightforward analogues of the above calculation, we arrive at the other formulae, namely:

$$\bar{\mathbf{N}}_{\eta\eta} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial \eta^2} - \left[\frac{\partial \bar{\Gamma}}{\partial \eta} \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \eta} \right] e_3 \quad (\text{B.4.39})$$

where

$$\begin{aligned}\frac{\partial^2 \bar{\Gamma}}{\partial \eta^2} &= \bar{R}_0^\top [\rho_{\eta\eta} \gamma_c + 2\rho_\eta e_2] \\ \frac{\partial \bar{\Gamma}}{\partial \eta} &= \bar{R}_0^\top [\rho_\eta \gamma_c + \rho_c e_2]\end{aligned}\tag{B.4.40}$$

and $\rho_{\xi\xi}$ can be obtained from the model equation (B.1.25).

Computing $\bar{\mathbf{N}}_{\xi\eta}$.

$$\bar{\mathbf{N}}_{\xi\eta} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial \eta \partial \xi} - \left[\frac{\partial \bar{\Gamma}}{\partial \eta} \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \right] e_3\tag{B.4.41}$$

where

$$\frac{\partial^2 \bar{\Gamma}}{\partial \eta \partial \xi} = \bar{R}_0^\top \frac{\partial^2 \Gamma}{\partial \eta \partial \xi} = \bar{R}_0^\top [\rho_{\xi\eta} \gamma_c + \rho_\eta e_1 + \rho_\xi e_2]\tag{B.4.42}$$

$$\frac{\partial \bar{\Gamma}}{\partial \xi} = \bar{R}_0^\top [\rho_\eta \gamma_c + \rho_c e_1]\tag{B.4.43}$$

$$\frac{\partial \bar{\Gamma}}{\partial \eta} = \bar{R}_0^\top [\rho_\eta \gamma_c + \rho_c e_2]\tag{B.4.44}$$

and $\rho_{\xi\eta}$ can be obtained from the model equation (B.1.25).

Time-derivatives of $\bar{\mathbf{N}}$. In order to take these derivatives, we keep ξ, η constant, and change t . Then ρ has to change, due to (B.1.25), and also \bar{x}, \bar{y} . We thus treat t similarly to ξ and η , taking caution with new time-dependent parameters that now have non-zero derivatives.

$$\bar{\mathbf{N}}_t = -k_1 \frac{\partial \bar{\Gamma}}{\partial \bar{x}} \frac{\partial \bar{x}}{\partial t} - k_2 \frac{\partial \bar{\Gamma}}{\partial \bar{y}} \frac{\partial \bar{y}}{\partial t}\tag{B.4.45}$$

Thus,

$$\bar{\mathbf{N}}_t = \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial t}\tag{B.4.46}$$

where:

$$\bar{\Gamma} = \bar{R}^\top (\Gamma - \bar{T})\tag{B.4.47}$$

$$\frac{\partial \bar{\Gamma}}{\partial t} = \frac{\partial \bar{R}^\top}{\partial t} (\Gamma - \bar{T}) + \bar{R}^\top \frac{\partial}{\partial t} (\Gamma - \bar{T})\tag{B.4.48}$$

$$= \frac{\partial \bar{R}^\top}{\partial t} (\Gamma - \bar{T}) + \bar{R}^\top \frac{\partial \Gamma}{\partial t} - \bar{R}^\top \frac{\partial \bar{T}}{\partial t}\tag{B.4.49}$$

and

$$\dot{\Gamma} = \dot{\rho} \gamma + \rho \dot{\gamma} = \dot{\rho} \gamma\tag{B.4.50}$$

$$\dot{\bar{R}} = \Omega_\times \bar{R}_0\tag{B.4.51}$$

$$\dot{\bar{T}} = \frac{\partial}{\partial t} [R(t) \rho_c(0) \gamma_c(0) + \mathcal{T}(t)]\tag{B.4.52}$$

$$= \Omega_\times \rho_c(0) \gamma_c(0) + \dot{\mathcal{T}}(t)\tag{B.4.53}$$

thus

$$\frac{\partial \bar{\Gamma}}{\partial t} = -\bar{R}_0^\top \Omega_\times (\mathbf{\Gamma} - \bar{T}) + \bar{R}^\top \rho_t \boldsymbol{\gamma} - \bar{R}^\top \Omega_\times \rho_c(0) \boldsymbol{\gamma}_c(0) - \bar{R}^\top \dot{T}(t) \quad (\text{B.4.54})$$

for $\bar{\Gamma} = t = 0$:

$$\frac{\partial \bar{\Gamma}}{\partial t} = \bar{R}_0^\top \left[\rho_t \boldsymbol{\gamma}_c(0) - \Omega_\times \rho_c(0) \boldsymbol{\gamma}_c(0) - \dot{T}(0) \right] \quad (\text{B.4.55})$$

The final formula for $\bar{x} = \bar{y} = t = 0$ is:

$$\bar{\mathbf{N}}_t = \mathbf{K} \bar{R}_0^\top \left[\rho_t \boldsymbol{\gamma}_c(0) - \Omega_\times \rho_c(0) \boldsymbol{\gamma}_c(0) - \dot{T}(0) \right] \quad (\text{B.4.56})$$

where $\frac{\partial \rho}{\partial t}$ can also be computed from the model equation (B.1.25). Similarly,

$$\bar{\mathbf{N}}_{tt} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial t^2} - \left[\frac{\partial \bar{\Gamma}^\top}{\partial t} \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial t} \right] e_3 \quad (\text{B.4.57})$$

where

$$\frac{\partial^2 \bar{\Gamma}}{\partial t^2} = \bar{R}_0^\top \left[\rho_{tt} \boldsymbol{\gamma}_c - 2\Omega_\times \rho_t \boldsymbol{\gamma}_c + 2\Omega_\times \Omega_\times \rho_c \boldsymbol{\gamma}_c + 2\Omega_\times \dot{T} \right] \quad (\text{B.4.58})$$

and $\ddot{\rho}$ can be obtained from the model equation (B.1.25). Similarly,

$$\bar{\mathbf{N}}_{\xi t} = \mathbf{K} \frac{\partial^2 \bar{\Gamma}}{\partial t \partial \xi} - \left[\frac{\partial \bar{\Gamma}^\top}{\partial t} \mathbf{K} \mathbf{K} \frac{\partial \bar{\Gamma}}{\partial \xi} \right] e_3 \quad (\text{B.4.59})$$

where

$$\frac{\partial^2 \bar{\Gamma}}{\partial t \partial \xi} = \frac{\partial}{\partial t} \left[\bar{R}^\top (\rho_\xi \boldsymbol{\gamma} + \rho e_1) \right] \quad (\text{B.4.60})$$

$$= \dot{\bar{R}}^\top (\rho_\xi \boldsymbol{\gamma} + \rho e_1) + \bar{R}^\top \frac{\partial}{\partial t} (\rho_\xi \boldsymbol{\gamma} + \rho e_1) \quad (\text{B.4.61})$$

$$= -\bar{R}_0^\top \Omega_\times (\rho_\xi \boldsymbol{\gamma} + \rho e_1) + \bar{R}^\top (\rho_{\xi t} \boldsymbol{\gamma} + \rho_t e_1) \quad (\text{B.4.62})$$

and $\rho_{\xi t}$ can be obtained from the model equation (B.1.25).

B.4.1 Derivatives of depth

Summary of formulas

$$\rho_\xi = \frac{\rho_c e_1^\top \bar{R} e_3}{\boldsymbol{\gamma}_c^\top \bar{R} e_3} \quad (\text{B.4.63})$$

$$\rho_\eta = \frac{\rho_c e_2^\top \bar{R} e_3}{\boldsymbol{\gamma}_c^\top \bar{R} e_3} \quad (\text{B.4.64})$$

Derivation

We start with the incremental version of (B.1.25)

$$(\gamma_c + \tilde{\gamma})^\top \mathbf{K}_c (\gamma_c + \tilde{\gamma}) \tilde{\rho}^2 + [(\gamma_c + \tilde{\gamma})^\top 2\mathbf{K}_c \tilde{\gamma} + (\gamma_c + \tilde{\gamma})^\top \bar{R}e_3] \tilde{\rho} + \tilde{\gamma}^\top \mathbf{K}_c \tilde{\gamma} \rho_c^2 + e_3^\top \bar{R}^\top \tilde{\gamma} \rho_c = 0 \quad (\text{B.4.65})$$

$$\gamma^\top \mathbf{K}_c \gamma \tilde{\rho}^2 + [\gamma^\top 2\mathbf{K}_c \tilde{\gamma} + \gamma^\top \bar{R}e_3] \tilde{\rho} + \tilde{\gamma}^\top \mathbf{K}_c \tilde{\gamma} \rho_c^2 + e_3^\top \bar{R}^\top \tilde{\gamma} \rho_c = 0 \quad (\text{B.4.66})$$

Note that the derivatives of $\tilde{\rho}$ are the same as the derivatives of ρ .

B.5 Coefficients of model in camera frame

In this supplementary section we list the coefficients h_i of planar flow (B.2.23) and of the paraboloid model (B.2.20).

Plane:

$$h_1 = (c_3 \Omega_x c_1 - c_3 \Omega_y c_2 - c_2^2 \Omega_z - c_1^2 \Omega_z) \bar{\mathcal{T}}_y(0) + \quad (\text{B.5.1})$$

$$(c_2 \Omega_z b_2 + c_3 \Omega_y b_2 + c_1 \Omega_z b_1 - c_3 \Omega_x b_1) \bar{\mathcal{T}}_z(0) + (b_2 c_1 - c_2 b_1) \dot{\bar{\mathcal{T}}}_z(0) \quad (\text{B.5.2})$$

$$h_2 = (c_3 \Omega_y c_2 - c_3 \Omega_x c_1 + c_2^2 \Omega_z + c_1^2 \Omega_z) \mathcal{T}_x(0) + \quad (\text{B.5.3})$$

$$(-a_2 \Omega_z c_2 + c_3 \Omega_x a_1 - c_3 \Omega_y a_2 - a_1 \Omega_z c_1) \mathcal{T}_z(0) + (c_2 a_1 - a_2 c_1) \dot{\mathcal{T}}_z(0) \quad (\text{B.5.4})$$

$$h_3 = (c_3 \Omega_x b_1 - c_2 \Omega_z b_2 - c_1 \Omega_z b_1 - c_3 \Omega_y b_2) \mathcal{T}_x(0) + \quad (\text{B.5.5})$$

$$(2 a_2 \Omega_z c_2 - c_3 \Omega_x a_1 + 2 a_1 \Omega_z c_1 - a_3 \Omega_x c_1 + c_3 \Omega_y a_2 + a_3 \Omega_y c_2) \mathcal{T}_y(0) + \quad (\text{B.5.6})$$

$$(-a_2 \Omega_z b_2 - a_1 \Omega_z b_1 - a_3 \Omega_y b_2 + a_3 \Omega_x b_1) \mathcal{T}_z(0) + \quad (\text{B.5.7})$$

$$(c_2 b_1 - b_2 c_1) \dot{\mathcal{T}}_x(0) + (-b_2 a_1 + a_2 b_1) \dot{\bar{\mathcal{T}}}_z(0) \quad (\text{B.5.8})$$

$$h_4 = (-a_3 \Omega_y c_2 - a_2 \Omega_z c_2 - a_1 \Omega_z c_1 + a_3 \Omega_x c_1) \mathcal{T}_x(0) + \quad (\text{B.5.9})$$

$$(a_1^2 \Omega_z + a_2^2 \Omega_z + a_3 \Omega_y a_2 - a_3 \Omega_x a_1) \mathcal{T}_z(0) + (a_2 c_1 - c_2 a_1) \dot{\bar{\mathcal{T}}}_x(0) \quad (\text{B.5.10})$$

$$h_5 = (a_3 \Omega_y b_2 + a_1 \Omega_z b_1 - a_3 \Omega_x b_1 + a_2 \Omega_z b_2) \mathcal{T}_x(0) + \quad (\text{B.5.11})$$

$$(-a_1^2 \Omega_z - a_2^2 \Omega_z - a_3 \Omega_y a_2 + a_3 \Omega_x a_1) \mathcal{T}_y(0) + (-a_2 b_1 + b_2 a_1) \dot{\bar{\mathcal{T}}}_x(0) \quad (\text{B.5.12})$$

$$h_6 = (c_2 b_1 - b_2 c_1) \mathcal{T}_x(0) + (a_2 c_1 - c_2 a_1) \mathcal{T}_y(0) + (-a_2 b_1 + b_2 a_1) \mathcal{T}_z(0) \quad (\text{B.5.13})$$

$$h_7 = (-b_3 \Omega_x c_1 + b_3 \Omega_y c_2 + c_2 \Omega_z b_2 + c_1 \Omega_z b_1) \mathcal{T}_y(0) + \quad (\text{B.5.14})$$

$$(-b_3 \Omega_y b_2 - b_1^2 \Omega_z - b_2^2 \Omega_z + b_3 \Omega_x b_1) \mathcal{T}_z(0) + (c_2 b_1 - c_1 b_2) \dot{\bar{\mathcal{T}}}_y(0) \quad (\text{B.5.15})$$

$$h_8 = (-2 c_1 \Omega_z b_1 - b_3 \Omega_y c_2 - 2 c_2 \Omega_z b_2 + b_3 \Omega_x c_1 + c_3 \Omega_x b_1 - c_3 \Omega_y b_2) \mathcal{T}_x(0) + \quad (\text{B.5.16})$$

$$(a_2 \Omega_z c_2 + a_1 \Omega_z c_1 + c_3 \Omega_y a_2 - c_3 \Omega_x a_1) \mathcal{T}_y(0) + \quad (\text{B.5.17})$$

$$(a_2 b_3 \Omega_y + a_2 \Omega_z b_2 - b_3 \Omega_x a_1 + a_1 \Omega_z b_1) \mathcal{T}_z(0) \quad (\text{B.5.18})$$

$$+ (a_2 c_1 - c_2 a_1) \dot{\bar{\mathcal{T}}}_y(0) + (-b_2 a_1 + a_2 b_1) \dot{\bar{\mathcal{T}}}_z(0) \quad (\text{B.5.19})$$

$$h_9 = (b_1^2 \Omega_z + b_3 \Omega_y b_2 + b_2^2 \Omega_z - b_3 \Omega_x b_1) \mathcal{T}_x(0) + \quad (\text{B.5.20})$$

$$(-a_2 b_3 \Omega_y - a_2 \Omega_z b_2 - a_1 \Omega_z b_1 + b_3 \Omega_x a_1) \mathcal{T}_y(0) + (-a_2 b_1 + b_2 a_1) \dot{\bar{\mathcal{T}}}_y(0) \quad (\text{B.5.21})$$

Paraboloid:

$$q_1 = a_1^2 \frac{k_1}{2} + a_2^2 \frac{k_2}{2} \quad (\text{of } x^2) \quad (\text{B.5.22})$$

$$q_2 = b_1^2 \frac{k_1}{2} + b_2^2 \frac{k_2}{2} \quad (\text{of } y^2) \quad (\text{B.5.23})$$

$$q_3 = c_1^2 \frac{k_1}{2} + c_2^2 \frac{k_2}{2} \quad (\text{of } z^2) \quad (\text{B.5.24})$$

$$q_4 = a_1 b_1 k_1 + a_2 b_2 k_2 \quad (\text{of } xy) \quad (\text{B.5.25})$$

$$q_5 = a_1 c_1 k_1 + a_2 c_2 k_2 \quad (\text{of } xz) \quad (\text{B.5.26})$$

$$q_6 = b_1 c_1 k_1 + b_2 c_2 k_2 \quad (\text{of } yz) \quad (\text{B.5.27})$$

$$q_7 = a_3 - x_c a_1^2 k_1 - x_c a_2^2 k_2 - y_c a_1 b_1 k_1 - y_c a_2 b_2 k_2 - z_c a_1 c_1 k_1 - z_c a_2 c_2 k_2 \quad (\text{of } x) \quad (\text{B.5.28})$$

$$q_8 = b_3 - x_c a_1 b_1 k_1 - x_c a_2 b_2 k_2 - y_c b_1^2 k_1 - y_c b_2^2 k_2 - z_c b_1 c_1 k_1 - z_c b_2 c_2 k_2 \quad (\text{of } y) \quad (\text{B.5.29})$$

$$q_9 = c_3 - x_c a_1 c_1 k_1 - x_c a_2 c_2 k_2 - y_c b_1 c_1 k_1 - y_c b_2 c_2 k_2 - z_c c_1^2 k_1 - z_c c_2^2 k_2 \quad (\text{of } z) \quad (\text{B.5.30})$$

$$q_{10} = q_1 x_c^2 + q_2 y_c^2 + q_3 z_c^2 + q_4 x_c y_c + q_5 x_c z_c + q_6 y_c z_c - a_3 x_c - b_3 y_c - c_3 z_c \quad (\text{const}) \quad (\text{B.5.31})$$

Appendix C

Bundle Adjustment

C.1 Introduction

Bundle adjustment is the process of estimating cameras and 3D points by minimizing reprojection error using a specialized iterative optimization scheme.

The description in [69] seems to be the most widely known introduction to the subject of bundle adjustment, and forms the basis for this section, while the main reference paper in the computer vision literature is [148]. The paper [41] also has a useful summary of the main concepts, as well as experiments showing that bundle adjustment can be used in real-time applications as in online video processing with great benefits. As such, it requires a good initialization, which can be challenging to compute for many views. The key to performance is to select the right number of frames to optimize at a time and the number of iterations to use in bundle adjustment. Even small amounts of bundle adjustment can be very beneficial in problems such as tracking over a large video sequence and solving for the relative pose of camera for each frame. Bundle adjustment is important not only for improving the accuracy of structure and motion solutions, but also to correct very harsh error-propagating effects that would otherwise render a method useless.

The Levenberg-Marquardt algorithm is the numeric method of choice for implementing bundle adjustment. It is a hybrid of Newton iteration and gradient descent, designed to provide faster convergence and regularization in the case of over-parametrized problems. In this text, we use the algorithm in several variables to provide least-squares solutions (rather than exact ones) to sets of equations. A good reference for Levenberg-Marquardt is [127].

C.2 Basic Formulation and Methods

Consider the problem of estimating a set of parameters $P \in \mathbb{R}^M$ from a set of observations $\bar{Z} \in \mathbb{R}^N$, $M \ll N$, typically, related by a model $Z = \mathcal{L}(P)$. The goal is to find P such that $Z(P)$ is as close to \bar{Z} as possible, *i.e.*, to minimize the residual prediction error $\|\Delta Z\| = \|\bar{Z} - Z(P)\|$. This problem

arises for example when a set of 3D points $\mathbf{\Gamma}_i^w$ have been reconstructed and camera parameters are sought for each view v in the form of \mathcal{R}_v , \mathcal{T}_v , and possibly intrinsic parameters \mathcal{K}_v . In this case the observations are the image feature points γ_{im} indexed by i in each view, $\bar{\gamma}_{im,i}^v$, *i.e.*,

$$\bar{Z} = [\bar{\gamma}_{im,i}^v], \quad i = 1, \dots, N_v, \quad v = 1, \dots, V. \quad (\text{C.2.1})$$

The model is a prediction in image coordinates of these features using a projection model $\gamma_i^v = \frac{\mathbf{\Gamma}_i^v}{\mathbf{e}_3^\top \mathbf{\Gamma}_i^v} = \frac{\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v}{\mathbf{e}_3^\top (\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v)}$ together with a translation into pixel coordinates $\gamma_{im,i}^v = \mathcal{K}_v \gamma_i^v$, *i.e.*,

$$Z(P) = [\gamma_{im,i}^v] = \left[\mathcal{K}_v \frac{\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v}{\mathbf{e}_3^\top (\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v)} \right], \quad (\text{C.2.2})$$

where $i = 1, \dots, N_v$ and $v = 1, \dots, V$. The prediction error requires a cost function $f(P, \bar{Z})$ which is typically in a more specific form of weighted sum of squares (SSE) error

$$f(P, \bar{Z}) = \|\bar{Z} - Z(P)\|^2 = \frac{1}{\bar{Z}} [\bar{Z} - Z(P)] W [\bar{Z} - Z(p)]. \quad (\text{C.2.3})$$

Specifically, the residual prediction error in this case is

$$\|\Delta Z\|^2 = \|\bar{Z} - Z(P)\|^2 = \left\| \left[\bar{\gamma}_{im,i}^v - \mathcal{K}_v \frac{\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v}{\mathbf{e}_3^\top (\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v)} \right] \right\|^2 = \sum_{i,v} \left\| \bar{\gamma}_{im,i}^v - \mathcal{K}_v \frac{\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v}{\mathbf{e}_3^\top (\mathcal{R}_v \mathbf{\Gamma}_i^w + \mathcal{T}_v)} \right\|^2, \quad (\text{C.2.4})$$

where $W = I$.

The computation of the optimal P , P^*

$$P^* = \underset{P}{\operatorname{argmin}} f(P, \bar{Z}) = \underset{P}{\operatorname{argmin}} \|\bar{Z} - Z(P)\|, \quad (\text{C.2.5})$$

is fairly simple if Z is linear in P , as this leads to a linear least squares solution. However, it is much more typical in multiview geometry problems that, due to projection, $Z(p)$ is nonlinear, *e.g.*, Equation C.2.4, thus requiring an approximate iterative solution from a good initial estimate P_0 . The intuition is that $Z(P)$ can be linearized or considered in quadratic form for each step from estimate P_i to P_{i+1} . In the linear, first-order approximation,

$$f(P + \Delta P) = f(P, \bar{Z}) + \nabla f^\top(P, \bar{Z}) \Delta P, \quad (\text{C.2.6})$$

where the steepest change is that of gradient descent

$$\lambda \Delta P = \nabla f(P, \bar{Z}), \quad (\text{C.2.7})$$

where $1/\lambda$ is the stepsize. For the SSE case this gives

$$\lambda \Delta P = -J^\top W^\top [\bar{Z} - Z(P)], \quad (\text{C.2.8})$$

where $J = \frac{dZ}{dP}$ is the Jacobian. The gradient descent quickly gets the estimate close to the minimum, but when it gets close, the gradient diminishes, and therefore the rate of convergence likewise decreases significantly. In this case a second-order approximation is required. Specifically, expanding $f(P, \bar{Z})$ in a Taylor expansion up to second order gives

$$f(P + \Delta P, \bar{Z}) = f(P, \bar{Z}) + \nabla f^\top(P, \bar{Z}) \Delta P + \frac{1}{2} \Delta P^\top H_f^\top(P, \bar{Z}) \Delta P, \quad (\text{C.2.9})$$

where ∇f and H_f are the gradient and the Hessian of f , respectively, with respect to P . Minimizing $f(P + \Delta P, \bar{Z})$ over ΔP gives

$$H_f(P, \bar{Z})\Delta P = -\nabla f(P, \bar{Z}). \quad (\text{C.2.10})$$

Thus, starting with P_0 , and iterating through

$$P_{i+1} = P_i + H_f^{-1}(P_i, \bar{Z})\nabla f(P_i, \bar{Z}), \quad i = 0, 1, 2, \dots \quad (\text{C.2.11})$$

is expected to converge to the solution P^* . This is the well-known Newton method.

Observe, however, that the computation of H_f requires the computation of not only the Jacobian $J = \frac{dZ}{dP}$, but also $\frac{d^2Z}{dP^2}$, which is not a trivial task. Fortunately, for the specific case of SSE in the form of Equation C.2.3 the Hessian can be approximated effectively using just the Jacobian J . Observe that in this case

$$\nabla f(P, \bar{Z}) = -[\bar{Z} - Z(P)]^\top W J, \quad (\text{C.2.12})$$

and

$$H_f(P, \bar{Z}) = J^\top W J - [\bar{Z} - Z(P)]^\top W \frac{d^2Z}{dP^2}. \quad (\text{C.2.13})$$

Now when $\bar{Z} - Z(p)$ is small or when $\frac{d^2Z}{dP^2}$ is negligible the second term can be ignored, leading to an approximate $\tilde{H}_f(P, \bar{Z})$,

$$\tilde{H}_f(P, \bar{Z}) = J^\top W J. \quad (\text{C.2.14})$$

In this case, Equation C.2.10 can be written in the form

$$J^\top W J \Delta P = -J^\top W (\bar{Z} - Z(P)). \quad (\text{C.2.15})$$

This is the Gaussian-Newton approximation, which is typically very effective in practice, and only requires the computation of the Jacobian J . In our case, $W = I$, so that

$$J^\top J \Delta P = -J^\top (\bar{Z} - Z(P)). \quad (\text{C.2.16})$$

The first-order gradient descent and the second-order Gauss-Newton approximations complement each other in that gradient descent gets the estimate close to the optimum but is ineffective in “finishing the job” where the second-order can do this. The two can be integrated by solving

$$(J^\top J + \lambda I)\Delta P = -J^\top [\bar{Z} - Z(P)], \quad (\text{C.2.17})$$

where $\lambda = 0$ gives Equation C.2.16 while for a high value of λ it gives Equation C.2.8. In this way the update is dominated by a first-order gradient descent initially and as the estimate approaches the minimum a second-order update is used. This procedure is referred to as **Levenberg-Marquardt**.

Different strategies are used to vary λ [69, 127]. Typically, the average of the diagonal of $J^\top J$ divided by 1000 is used to set λ , so that λ dominates initially. If the resulting ΔP reduces errors, then λ is divided by 10 for the next iteration; otherwise, λ is multiplied by 10 and the procedure is repeated until the cost function decreases. Therefore, it transitions smoothly from a Gauss-Newton iteration to gradient descent and back depending on which method is more effective.

In practice, one provides a routine to compute the cost f , a goal vector of observed or desired values, and an initial estimate of P_0 . The computation of the Jacobian can be done either numerically or by providing a routine. Numerical computation usually works well.

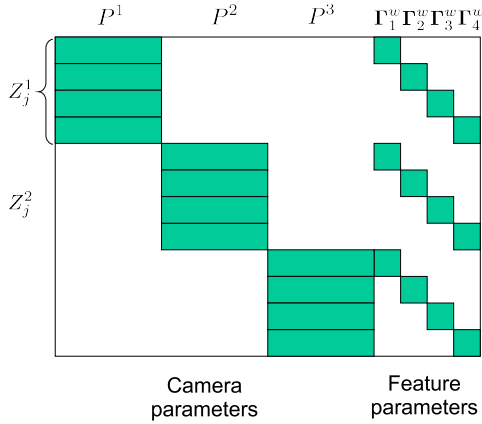


Figure C.1: Form of the Jacobian matrix for bundle adjustment estimating cameras and 3D points.

Sparse Levenberg-Marquardt: The bottleneck in the basic Levenberg-Marquardt algorithm is solving for ΔP in Equation C.2.17 which in general requires $O(N^3)$ operations, N being the number of parameters. However, in many problems such as the optimization of cameras and 3D points, the matrix $J^\top J$ has a sparse block structure. For example, the image of a point does not depend on any other point such that the “measurement vector” $\bar{Z} \in \mathbf{R}^N$ may be broken into $\bar{Z} = (\bar{Z}_1^\top, \dots, \bar{Z}_n^\top)^\top$, and the parameter vector may be broken into $P = (a^\top, b_1^\top, \dots, b_n^\top)^\top$, a sparseness assumption is that each Z_i is dependent on a and b_i only, i.e., $\frac{\partial Z_i}{\partial b_j} = 0$, for $i \neq j$. In the camera and point reconstruction problem, b_i is the vector of parameters of the i -th point, and Z_i the vector of projections of this point in all the views, which does not depend on any other point being estimated. The form of the Jacobian matrix is as in Figure C.1, where Z_i^j denotes point i projected in view j .

The Jacobian has block structure

$$J = [A|B_1|B_2|\dots|B_n], \quad \text{where } A = \begin{bmatrix} \partial Z \\ \partial a \end{bmatrix}, \quad B_i = \begin{bmatrix} \partial Z \\ \partial b_i \end{bmatrix}. \quad (\text{C.2.18})$$

The equations to compute the update on parameters ΔP now have the form

$$J\Delta P = [A|B] \begin{pmatrix} \Delta_a P \\ \Delta_b P \end{pmatrix} = \Delta Z. \quad (\text{C.2.19})$$

Other sources of sparseness are (i) the lack of interaction between parameters of different cameras under proper formulation of the problem, and (ii) correspondences are not always visible in all views. Each step of the sparse Levenberg-Marquardt implementation is $O(N)$ in the number of parameters. Hartley and Zisserman [69] list the algorithm in detail.

Large-scale bundle adjustment: Recently, authors have applied bundle adjustment for reconstructing cameras and 3D points for entire cities, comprising almost a million cameras. In this case, instead of sparse Levenberg-Marquardt, some authors have resorted to the conjugate gradient method in order to optimize all cameras and all points simultaneously. This approach allows for optimizing a very large system of equations, at the cost of being slower than Levenberg-Marquardt.

Iterative Fundamental Matrix Estimation: There is a method for fundamental matrix estimation using Levenberg-Marquardt. It basically is the multiview bundle adjustment algorithm, but for pairs of cameras. The first camera is assumed to be $[I|0]$, and the second one is estimated iteratively, together with 3D points. The result is cameras and points up to projective ambiguity, but this is equivalent to the fundamental matrix. If the second camera has matrix $[M|m]$, then the fundamental matrix is $[m]_{\times}M$. This is sometimes called bundle adjustment, although this is not strict terminology since a specialized sparse implementation of Levenberg-Marquardt is not required.

Appendix D

Geometric and Qualitative Approach to Epipolar Geometry

This chapter presents material from ongoing research on qualitative and geometric aspects of epipolar geometry estimation for use in automatic structure from motion systems. It has more of an experimental and open-ended flavor than the rest of this thesis, so it was made into an appendix.

D.1 Introduction

D.1.1 Motivation and Outline

Reconstruction of 3D structure and cameras from a number of 3D views has always been one of the central problems in computer vision, called structure from motion, and is the major topic of this thesis. A burst of research has occurred since the 90's, when projective methods enabled reconstruction with little prior on the intrinsic parameters. Moreover, improved interest point detectors and local descriptors more recently enabled wider application of reconstruction systems without need to first solve the segmentation problem. These advances have enabled highly flexible systems, but this comes at the price of poor accuracy of the results (for traditional photogrammetric standards), due to instability when dealing with noisy input [118] under weak assumptions. Moreover, in some cases there might be a lack of sufficient point features for a stable camera estimation, but there is still enough curve content which hasn't been exploited by state-of-the-art systems.

Given features detected in two images, current systems for automatic 3D calibration and reconstruction usually consist in:

- i) Initial matching of sparse features to bootstrap the system
- ii) Simultaneous recovery epipolar geometry and consistent correspondences
- iii) Projective reconstruction of cameras and sparse features,

- iv) Auto-calibration and Euclidean reconstruction,
- v) Dense matching and reconstruction.

Epipolar geometry is the image-based representation of the relative camera pose and intrinsic parameters between two views, and is the most important tool in all steps (ii–v). In this chapter, a new geometric representation for epipolar geometry is proposed and studied, and a new qualitative approach for its estimation is explored. We will focus on its use for step (ii): given interest points detected in two images and initial correspondences containing a high percentage of outliers, the output is an epipolar geometry most consistent with the data, and a subset of initial correspondences consistent with the epipolar geometry.

The traditional approach for robust estimation of epipolar geometry is based on RANSAC [54, 69] and feature points such as Harris corners or SIFT. Basically, we start from an initial set of rough matches between two views, which are obtained by some similarity measure of the appearance around the feature. We then pick different subsets of correspondences (as small as possible), compute an epipolar geometry for each, and keep the result which is most consistent with all the initial correspondences. The key aspects of the system are: the number of correspondences in each subset, the core algorithm to use for instantiating the epipolar geometry for a given subset, how many subsets to try, and how to validate a particular subset. Each of these are essentially dictated by the dataset (*e.g.* proportion of outliers, localization noise) and the core algorithm used within RANSAC (*e.g.* number of points required, stability). In many applications we have little control over the data, such as when processing videos from the internet, so we should do the best possible in choosing a core algorithm, and our contribution is in this direction.

The ideas presented in this chapter have a long-term goal to serve as a foundation for future research on estimating epipolar geometry from curves. The qualitative aspects exploited here can be readily applied to curves, generating aspect graphs of solutions as first suggested by Porill and Pollard [126]. Our research with such aspect graphs for the case of points can help identify the feasibility of implementing such idea for the case of curves.

D.1.2 Fundamental Concepts of Epipolar Geometry

Epipolar geometry arises as an answer to the following question: Given two projected points p^1 and p^2 in two views, can they correspond to the same 3D point? Let us define some basic concepts.

Definition 18. (*Basic geometric entities*)

- The *epipolar line* el_1^2 of a point p^1 in image 2, or $el_1^2(p^1)$ is the projection in image 2 of the line in space of all 3D points that project to p^1 . We define $el_2^1(p^2)$ in a similar way.
- The *epipole* e_1^2 of camera 1 on image 2 is the projection in image 2 of the camera center \mathbf{c}_1 . We define e_2^1 in a similar way.
- The *epipolar plane* of a 3D point Γ^w is the plane formed by $\mathbf{c}_1, \mathbf{c}_2, \Gamma^w$.

Property 1. All epipolar lines $el_1^2(p^1)$ necessarily pass through the epipole e_1^2 .

Proof. The set of 3D points projecting to p^1 defines a straight line that has to include c_1 . Thus, projecting all points of this line in image 2 includes projecting c_1 . ■

Property 2. The intersection of the epipolar plane of Γ^w with one image plane is a line $el_2^1(el_1^2)$, which is called the epipolar line of Γ^w , which is the epipolar line of $p^1(p^2)$.

Proof. The point p^1 is in el_2^1 and also in the epipolar plane. The epipole is in el_2^1 and also in the epipolar plane. Thus, el_2^1 must be the intersection of the epipolar plane and the image plane. ■

The epipolar geometry is algebraically encoded by the fundamental matrix in the (intrinsically) uncalibrated case, and by the Essential matrix in the calibrated case, as reviewed in the sections to follow. The Euclidean-geometric concepts, however, are not immediately clear from the projective geometry. Instead of matrices, we could represent epipolar geometry more explicitly by two epipoles and a mapping between a set of corresponding epipolar lines. The following questions arise: What is the advantage of an explicit geometric representation? How do we know this is the right representation to be used? For instance, epipoles may be at infinity; this should be treated in an explicit geometric formulation. How to go from an essential matrix or fundamental matrix algebraic representation to the geometric representation and vice-versa? In this chapter we investigate such geometric representation and explore geometric interpretations to many aspects of epipolar geometry that is usually encoded quite implicitly in the algebra.

D.1.3 The Essential Matrix

Let Γ^1 be the non-homogeneous coordinates of a 3D point Γ in the 3D frame of camera 1, and Γ^2 the coordinates of the same point in camera 2. Then we can write, for some rotation \mathcal{R} and translation \mathcal{T} :

$$\Gamma^2 = R\Gamma^1 + \mathcal{T} \quad (\text{D.1.1})$$

Taking the cross-product with \mathcal{T} on both sides, we get:

$$\mathcal{T} \times \Gamma^2 = \mathcal{T} \times R\Gamma^1 \quad (\text{D.1.2})$$

taking the scalar product with Γ^2 ,

$$\Gamma^2 \cdot (\mathcal{T} \times R\Gamma^1) = 0 \quad (\text{D.1.3})$$

$$\Gamma^{2\top} (\mathcal{T}_\times R) \Gamma^1 = 0 \quad (\text{D.1.4})$$

where R is a rotation matrix, and \mathcal{T}_\times is a skew-symmetric matrix formed from the translation vector such that $\mathcal{T}_\times \Gamma = \mathcal{T} \times \Gamma$. The essential matrix is then given by:

$$E = \mathcal{T}_\times R \quad (\text{D.1.5})$$

up to a scale factor (since Equation D.1.4 is equal to zero, the scale factor makes no difference). Now, if instead of the 3D point coordinates $\mathbf{\Gamma}^1$ and $\mathbf{\Gamma}^2$ we have the 2D point coordinates $\boldsymbol{\gamma}^i = (\xi^i, \eta^i, 1)^\top$, $i = 1, 2$, we can substitute $\mathbf{\Gamma}^i = \rho^i \boldsymbol{\gamma}^i$ into equation (D.1.4) to get

$$\boldsymbol{\gamma}^{2\top} E \boldsymbol{\gamma}^1 = 0, \quad \text{where } E = T_\times R. \quad (\text{D.1.6})$$

Some questions arise: When we know E , what constraint does $E = T_\times R$ place on T or R separately? What does this mean geometrically? How to arrive at the fact that this is exactly equivalent to having two epipoles and a map between epipolar lines, *i.e.*, how do we know that E fully represents the geometry of the problem?

Epipoles and the Essential Matrix. Given an essential matrix E , the epipole e_2^1 is defined as the point in image 1 satisfying the epipolar constraint for all points in image 2:

$$\boldsymbol{\gamma}^{2\top} E e_2^1 = 0, \quad \forall \boldsymbol{\gamma}^2 \text{ in image 2,} \quad (\text{D.1.7})$$

resulting in

$$E e_2^1 = 0, \quad (\text{D.1.8})$$

i.e., e_2^1 is algebraically given by the right null-vector of E (in homogeneous coordinates). Similarly, it can be shown that e_1^2 is the left null-vector of E .

Interpreting the Essential Matrix First, note that the translation T is the coordinates of point \mathbf{c}_1 in camera 2. To see this, we know that \mathbf{c}_1 is the origin of the coordinate system of camera 1, so that $\mathbf{\Gamma}^1 = (0, 0, 0)^\top$ at \mathbf{c}_1 . Since $\mathbf{\Gamma}^2 = R\mathbf{\Gamma}^1 + T$, we have that $\mathbf{\Gamma}^2 = T$ for the point \mathbf{c}_1 .

Second, the vector $R\boldsymbol{\gamma}^1$ is just the vector from \mathbf{c}_1 to the image point $\boldsymbol{\gamma}^1$, but written in coordinates of camera 2, as shown in Figure D.1. Therefore, $T \times R\boldsymbol{\gamma}^1$ is a vector that is normal to the epipolar plane spanned by T and the 3D point $\mathbf{\Gamma}^2$, all written in coordinates of camera 2. Thus, the equation $\boldsymbol{\gamma}^{2\top} \cdot T \times R\boldsymbol{\gamma}^1$ means that the vector $\boldsymbol{\gamma}^2$ has to lie in the epipolar plane. See Figure D.1. Since we also have the restriction that $\boldsymbol{\gamma}^2$ is in the second image plane, we have that it must lie in the intersection of two planes, and, thus, there is an epipolar line in the second image consisting of all those image points $\boldsymbol{\gamma}^2$ that satisfy the epipolar constraint.

Suppose we don't know that $\mathbf{\Gamma}^1$ and $\mathbf{\Gamma}^2$ are coordinates of the same 3D point, but we know they satisfy the essential constraint of Equation (D.1.4). This means that their projection lines intersect in 3D, but does not mean that they are 3D coordinates of the same point, of course. Note, however, that two coplanar lines intersect always, except when they are parallel. The parallel case has to be given special treatment if we use non-homogeneous coordinates.

Epipolar Constraint in World Coordinates. In our previous paper [46], all vectors were in 3D world coordinates, and instead of attaching a 3D coordinate frame to each camera, we had the focal vector, which models two rotation parameters, but not camera roll. Since only world

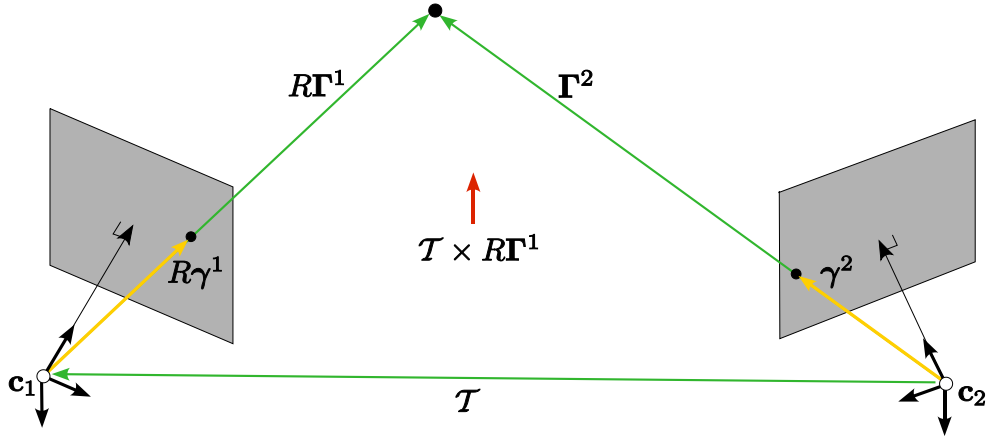


Figure D.1: Interpretation of the essential matrix equation. All vectors are written with respect to the 3D coordinate system of camera 2.

coordinates are being used, the epipolar constraint is even simpler and this gives insight into the essential matrix. Let us start by relating the world point Γ^w to each image point:

$$\rho^1 \gamma^1 + \mathbf{c}_1 = \rho^2 \gamma^2 + \mathbf{c}_2 = \Gamma^w \quad (\text{D.1.9})$$

where Γ^w represents the world coordinates of the 3D point.

$$\rho^1 \gamma^1 = \rho^2 \gamma^2 + \mathbf{c}_2 - \mathbf{c}_1 \quad (\text{D.1.10})$$

$$(\mathbf{c}_2 - \mathbf{c}_1) \times \rho^1 \gamma^1 = (\mathbf{c}_2 - \mathbf{c}_1) \times \rho^2 \gamma^2 \quad (\text{D.1.11})$$

$$\gamma_2 \cdot [(\mathbf{c}_2 - \mathbf{c}_1) \times \rho^1 \gamma^1] = 0 \quad (\text{D.1.12})$$

$$\gamma_2^\top (\mathbf{c}_2 - \mathbf{c}_1) \times \gamma^1 = 0 \quad (\text{D.1.13})$$

the Essential matrix is then

$$E = (\mathbf{c}_2 - \mathbf{c}_1)_\times \quad (\text{D.1.14})$$

Thus, an epipolar constraint involves the knowledge of a normal to the epipolar plane, *e.g.* the epipolar plane of the given point in image 1. One component of this normal depends on the point. The other component, however, only depends on the cameras, not on any point. It is such camera-dependent component that is represented by the essential matrix.

D.1.4 The Fundamental Matrix

The Fundamental matrix is the analog of the Essential matrix for operating directly in image coordinates, and it becomes useful when the intrinsic parameters of the cameras are unknown. The fundamental constraint for epipolar geometry is defined by the equation:

$$\gamma_{im}^{2\top} F \gamma_{im}^1 = 0, \quad (\text{D.1.15})$$

where γ_{im}^i , $i = 1, 2$ are the image points measured in image pixel coordinates and having third coordinate 1. The fundamental matrix is given by

$$F = K_2^{-\top} \mathcal{T}_\times \mathcal{R} K_1^{-1}, \quad (\text{D.1.16})$$

where K_1 and K_2 are the intrinsic parameter matrices of views 1 and 2, respectively. It is a 3×3 matrix with rank 2.

D.1.5 Literature Overview

In 1981, Longuet-Higgins [99] devised the essential matrix. Faugeras, Luong and others [52, 102, 106] devised the fundamental matrix in 1992, enabling the practical use of epipolar geometry for intrinsically uncalibrated cameras. This is considered a major milestone in Computer Vision.

Methods for estimating the Fundamental matrix are traditionally feature-based [69], even for infinitesimal motion [41]. More recently, however, a direct method (i.e. dense) for computing the fundamental matrix under infinitesimal motion has been proposed [137], which can improve results in scenes where salient features are absent, such as blurred images or textured images with aliasing. More information on infinitesimal epipolar geometry can be found at [104, 106, 151].

The idea behind the epipolar ordering constraint exploited in Section D.3 first appeared in [103], although we arrived at this constraint prior to knowing about this reference.

D.1.6 Classic Epipolar Geometry Estimation

The Normalized 8-point Algorithm

Outline. The 8 point algorithm involves the least-squares solution of a linear system of equations assuming the fundamental matrix as having general entries. The result is outside the space of fundamental matrices (which is the space of rank-2 matrices), so the algorithm finishes by projecting the solution back to this space. The data points have to be normalized and scaled such that the centroid of the reference points is at the origin of image coordinates, and the RMS distance of the points from the origin is equal to $\sqrt{2}$. Normalization is necessary to improve the conditioning of the underlying linear system, thus improving the stability of the estimation.

By giving sufficiently many point matches γ_i, γ'_i , we can compute the unknown matrix F from the fundamental constraint, Equation D.1.15. Each pair of matching points gives one equation in the entries of F :

$$\xi' \xi f_{11} + \xi' \eta f_{12} + \xi' f_{13} + \eta' \xi f_{21} + \eta' \eta f_{22} + \eta' f_{23} + \xi f_{31} + \eta f_{32} + f_{33} = 0. \quad (\text{D.1.17})$$

Denote by \mathbf{f} the 9-vector made up of the entries of F in row-major order. Then we can express the equation as:

$$(\xi' \xi, \xi' \eta, \xi', \eta' \xi, \eta' \eta, \eta', \xi, \eta, 1) \mathbf{f} = 0. \quad (\text{D.1.18})$$

Substituting into the equation $\det(F) = 0$ gives rise to a homogeneous cubic equation in the variables μ and ν (or a non-homogeneous cubic equation in a single parameter). There will be 3 solutions in general, but the complex ones are discarded [68].

Less than 7 points

In this case the problem becomes underconstrained. Hesse [75] showed that if 6 point correspondences are given, then the epipoles each lie on a cubic plane curve.

D.2 Geometric Approach to Epipolar Geometry

“In the absence of a geometric interpretation, a large set of polynomial equations can be about as informative as a block of assembler code” – Stephen Maybank

The Fundamental matrix has 7 DOF (degrees of freedom), which are described in the literature as 4 DOF for the epipoles and 3 for the transformation between epipolar lines, which is shown to be a 1D homography [51, 69]. However, the arguments for this are mostly algebraic, with little geometric insight on *how* the 7 DOF relate to the geometry of the two cameras, or the question of how is it that only two epipoles and three corresponding epipolar lines are enough to specify the complete epipolar geometry. Another question that motivated us is: if we know the fundamental matrix, what do we know about the cameras? The literature tells us that there is a fundamental ambiguity, where a set of projectively related cameras are all consistent with the same fundamental matrix. However, the proof to this theorem is algebraic, and we wish to augment the classic knowledge with more geometric insight.

Let us represent the epipolar geometry by two epipoles e^1 and e^2 , a pencil of epipolar lines through each epipole, and a mapping between the two pencils. An epipolar pencil can be represented by a family of lines which, under a given planar coordinate system having the epipole as origin, have equation

$$\eta = \tan \psi \xi \tag{D.2.1}$$

where ψ is the angle of the particular line with the ξ axis, so that the pencil is parametrized by ψ .

D.2.1 Effect of Planar Transformations on Epipolar Geometry

The intrinsic parameters and in-plane rotation of an image form an affine group of transformations on the coordinate system on an image plane. Let us analyze how each type of transformation affects the epipolar geometry. This study sheds light on the relationship between epipolar geometry and camera parameters. In what follows, we have a coordinate system of a plane, indicated by ξ, η . The transformed coordinates are indicated by $\bar{\xi}, \bar{\eta}$. Later on, ξ, η will be given an interpretation, such as pixel coordinates, but, for now, there is no specific interpretation.

Scaling. Scaling of the coordinate system in two orthogonal directions is specified by σ_ξ and σ_η . The relationship between these factors and the usual intrinsic parameters in the literature will be described further in this section. Assuming the epipole is at the origin, the scaling transformation is written as:

$$\begin{cases} \bar{\xi} = \sigma_\xi \xi \\ \bar{\eta} = \sigma_\eta \eta. \end{cases} \quad (\text{D.2.2})$$

Using $\eta = \tan \psi \xi$,

$$\bar{\eta} = \sigma_\eta \tan \psi \xi \quad (\text{D.2.3})$$

$$\bar{\eta} = \frac{\sigma_\eta}{\sigma_\xi} \tan \psi \bar{\xi}. \quad (\text{D.2.4})$$

Therefore, the epipolar pencil in the scaled coordinates is given by:

$$\bar{\eta} = \tan \bar{\psi} \bar{\xi} \quad (\text{D.2.5})$$

where

$$\tan \bar{\psi} = \frac{\sigma_\eta}{\sigma_\xi} \tan \psi. \quad (\text{D.2.6})$$

Note that knowledge of σ_ξ or σ_η in isolation is not important, only their *ratio* is relevant. Moreover, vertical and horizontal lines remain the same under scaling.

Translation. Translation occurs in practice when the epipole is given in pixels, while we need to relate this to the camera coordinates in world units relative to the principal point. Thus, let's say that the epipole coordinates were changed from ξ_0, η_0 to $\bar{\xi}_0, \bar{\eta}_0$. The angles under the new coordinates remain the same, since

$$\eta - \eta_0 = \tan \psi (\xi - \xi_0) \quad (\text{D.2.7})$$

and, using $\bar{\eta} = \eta + \bar{\eta}_0 - \eta_0$ and $\bar{\xi} = \xi + \bar{\xi}_0 - \xi_0$, we have

$$\bar{\eta} - \bar{\eta}_0 = \tan \psi (\bar{\xi} - \bar{\xi}_0). \quad (\text{D.2.8})$$

This result is also valid for vertical lines.

Skew. As illustrated in Figure D.2, we can write:

$$\begin{cases} \xi = \bar{\xi} + \bar{\eta} \cos \theta \\ \eta = \bar{\eta} \sin \theta. \end{cases} \quad (\text{D.2.9})$$

The inverse transformation is:

$$\begin{cases} \bar{\xi} = \xi - \eta \cot \theta \\ \bar{\eta} = \eta \csc \theta. \end{cases} \quad (\text{D.2.10})$$

Since we also wrote $\eta = \tan \psi \xi$, we can substitute the equations for $\eta(\bar{\xi}, \bar{\eta})$ and $\xi(\bar{\xi}, \bar{\eta})$ above to get:

$$\bar{\eta} = \frac{\tan \psi}{\sin \theta - \cos \theta \tan \psi} \bar{\xi} \quad (\text{D.2.11})$$

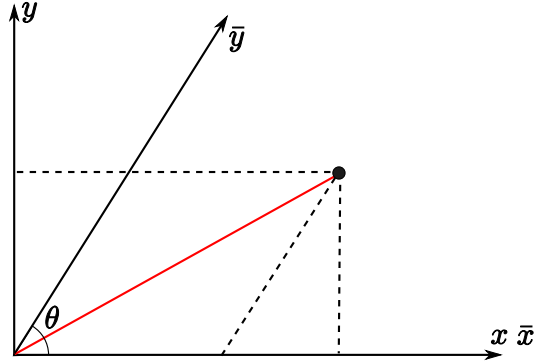


Figure D.2: Skewed image coordinates

Skew, translation, and scaling all together. The order of transformations used in practice is first scaling, then skew, then translation. The following proposition gives the formula for the entities θ , σ_ξ , σ_η , ξ_0 , η_0 describing such transformation, which we call the geometric intrinsic parameters.

Proposition D.2.1. (Geometric intrinsic parameters and epipolar lines) *Given a change of coordinates of the image plane consisting of, in order, scaling, skew, and translation:*

$$\begin{cases} \bar{\xi} = \sigma_\xi \xi + \sigma_\eta \cot \theta \eta + \bar{\xi}_0 \\ \bar{\eta} = \sigma_\eta \csc \theta \eta + \bar{\eta}_0, \end{cases} \quad (\text{D.2.12})$$

then the equation of a fixed pencil of lines around $e = (\xi_e, \eta_e)$ transforms from $\eta - \eta_e = \tan \psi (\xi - \xi_e)$ to $\bar{\eta} - \bar{\eta}_e = \tan \bar{\psi} (\bar{\xi} - \bar{\xi}_e)$, where $\tan \bar{\psi}$ is given by:

$$\tan \bar{\psi} = \frac{\sigma_\eta \tan \psi}{\sigma_\xi \sin \theta - \sigma_\eta \cos \theta \tan \psi} = \frac{\tan \psi}{\frac{\sigma_\xi}{\sigma_\eta} \sin \theta - \cos \theta \tan \psi}, \quad (\text{D.2.13})$$

and $(\bar{\xi}_e, \bar{\eta}_e)$ are the transformed coordinates of e .

Proof. It is clear that translation does not affect the angle of the epipolar lines, see Equation D.2.8. Scaling affects the angles as:

$$\tan \tilde{\psi} = \frac{\sigma_\eta}{\sigma_\xi} \tan \psi, \quad (\text{D.2.14})$$

where $\tilde{\psi}$ are the angles in the scaled coordinates, as shown in Equation D.2.6. A change from the scaled coordinates to the skewed coordinates then gives:

$$\tan \bar{\psi} = \frac{\tan \tilde{\psi}}{\sin \theta - \cos \theta \tan \tilde{\psi}}, \quad (\text{D.2.15})$$

as shown in Equation D.2.11. Plugging $\tan \tilde{\psi}$ from (D.2.14) into (D.2.15), we obtain the desired result. ■

Note that we could first skew, then scale, but this would be different than what is used in practice as described in Chapter 3.2. The following corollary adapts the result to the parameters used in the literature, which we call classic intrinsic parameters.

Corollary D.2.2. (Classic intrinsic parameters and epipolar lines). *Given a change of coordinates of the image plane consisting of, in order, scaling, skew, and translation:*

$$\begin{cases} \bar{\xi} = \alpha_\xi \xi + s_\theta \eta + \bar{\xi}_0 \\ \bar{\eta} = \alpha_\eta \eta + \bar{\eta}_0, \end{cases} \quad (\text{D.2.16})$$

or, in the classic form using homogeneous coordinates,

$$\begin{bmatrix} \bar{\xi} \\ \bar{\eta} \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_\xi & s_\theta & \bar{\xi}_0 \\ 0 & \alpha_\eta & \bar{\eta}_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \\ 1 \end{bmatrix} \quad (\text{D.2.17})$$

then the equation of a fixed pencil of lines around $e = (\xi_e, \eta_e)$ transforms from $\eta - \eta_e = \tan \psi (\xi - \xi_e)$ to $\bar{\eta} - \bar{\eta}_e = \tan \bar{\psi} (\bar{\xi} - \bar{\xi}_e)$, where $\tan \bar{\psi}$ is given by:

$$\tan \bar{\psi} = \frac{\alpha_\eta \tan \psi}{\alpha_\xi + s_\theta \tan \psi} = \frac{\tan \psi}{\frac{\alpha_\xi}{\alpha_\eta} + \frac{s_\theta}{\alpha_\eta} \tan \psi}, \quad (\text{D.2.18})$$

and $(\bar{\xi}_e, \bar{\eta}_e)$ are the transformed coordinates of the e .

Proof. The Equations (D.2.16) and (D.2.12) describe the same transformation. Therefore, the coefficients of ξ and η must equal:

$$\alpha_\xi = \sigma_\xi \quad (\text{D.2.19})$$

$$\alpha_\eta = \sigma_\eta \csc \theta \quad (\text{D.2.20})$$

$$s_\theta = \sigma_\eta \cot \theta. \quad (\text{D.2.21})$$

Our purpose is to obtain the factors $\frac{\sigma_\xi}{\sigma_\eta}$ and $\cos \theta$ and plug them into Equation (D.2.13). Let us write:

$$\sigma_\xi = \alpha_\xi \quad (\text{D.2.22})$$

$$\sigma_\eta = \alpha_\eta \sin \theta \quad (\text{D.2.23})$$

$$\cot \theta = \frac{s_\theta}{\sigma_\eta} \quad (\text{D.2.24})$$

Then, after some manipulation,

$$\frac{\sigma_\xi}{\sigma_\eta} = \frac{1}{\sin \theta} \frac{\alpha_\xi}{\alpha_\eta} \quad (\text{D.2.25})$$

$$\cos \theta = \frac{s_\theta}{\alpha_\eta \sin \theta} \quad (\text{D.2.26})$$

Plugging these into (D.2.13), after a trivial manipulation, the result follows. ■

The aforementioned corollary shows that, actually, knowledge of α_ξ , α_η , and s_θ in isolation is not necessary for the transformation of epipolar lines realized by the intrinsic parameters. Instead of these 3 numbers, only two ratios $\frac{\alpha_\xi}{\alpha_\eta}$ and $\frac{s_\theta}{\alpha_\eta}$ are needed. Therefore *only 4 DOF from the intrinsic camera parameters affect the epipolar geometry.*

In-plane rotation. Although this occurs in practice as one of the extrinsic parameters, it is fundamentally a 2D transformation, so we can attempt to treat it separately from the other extrinsic parameters.

D.2.2 Basic Geometric Aspects of Epipolar Homography

This section is concerned with the following questions: How does the map between epipolar lines look like? What is its most elementary nature? We shall prove it is a homography. Moreover, how does this map relate to the geometry of the cameras?

Theorem D.2.3. *The mapping between epipolar lines is a 1D homography. More precisely, let the epipolar geometry is represented by two epipoles \mathbf{e}_1 and \mathbf{e}_2 and epipolar lines are given by angles α_1 and α_2 around the respective epipoles. If directionality is not important, i.e., only the tangents $x = \tan \alpha_1$ and $y = \tan \alpha_2$ matter, then:*

$$y = \frac{ax + b}{cx + d}, \quad (\text{D.2.27})$$

where a, b, c, d are arbitrary coefficients.

The coefficients a, b, c, d can be written with respect to the geometry of the camera system as indicated in a subsequent Lemma.

Corollary D.2.4. *The relationship between $x = \tan \alpha_1$ and $y = \tan \alpha_2$ can be given symmetrically by an **alignment hyperbola**:*

$$axy + bx + cy + d = 0, \quad (\text{D.2.28})$$

where a, b, c , and d are arbitrary coefficients.

Lemma D.2.5. (Epipolar homography in terms of geometric parameters) *The mapping between epipolar lines is a 1D homography whose 3 DOF can be written as three angles which describe the relative geometry of an image plane relative to the first, namely two angles θ_2, ϕ_2 describing the normal of the second plane, and one angle θ_1 describing the direction of the second epipole with respect to the normal of the first plane. If we write the 1D epipolar homography in the form:*

$$y = \frac{ax + b}{cx + d}, \quad (\text{D.2.29})$$

then we can write the coefficients a, b, c and d in terms of the three geometric parameters of the system:

$$a = \sin \theta_1 \sin \theta_2 \cos \phi_2 (1 - \sin^2 \theta_2 \sin^2 \phi_2) + (\cos \theta_1 \cos \theta_2) (1 - \sin^2 \theta_2 \sin^2 \phi_2) - \sin \theta_1 \sin^3 \theta_2 \sin^2 \phi_2 - \cos \theta_1 \sin^2 \theta_2 \cos \theta_2 \sin^2 \phi_2 \quad (\text{D.2.30})$$

$$b = 2 \cos \theta_1 \cos \theta_2 \sin^2 \theta_2 \sin \phi_2 \cos \phi_2 \quad (\text{D.2.31})$$

$$c = \sin \phi_2 \sin \theta_2 (\cos \theta_1 \sin \theta_2 \cos \phi_2 - \sin \theta_1 \cos \theta_2) \quad (\text{D.2.32})$$

$$d = \cos \theta_1 (1 - \sin^2 \theta_2 \sin^2 \phi_2). \quad (\text{D.2.33})$$

Proof. Let the two image planes be described simply as two general planes π_1 and π_2 . An outline of the proof is as follows:

1. The epipole \mathbf{e}_1 is any point in the plane π_1 .
2. The unit translation direction \mathbf{T}_1 is any direction forming angle θ_1 with the normal \mathbf{N}_1 . This direction points to the epipole \mathbf{e}_2 in the second plane. There is a cone of vectors, but this is not important for the intrinsic configuration of two planes. So \mathbf{T}_1 is specified by the angle θ_1 :

$$\mathbf{T}_1 \cdot \mathbf{N}_1 = \cos \theta_1 \quad (\text{D.2.34})$$

while one DOF ϕ_1 is not specified for this proof.

3. We now have a universal coordinate frame formed by the vectors \mathbf{N}_1 , \mathbf{T}_1^\perp , and $\mathbf{T}_1^\perp \times \mathbf{N}_1$. Lets choose the ordering $[\mathbf{T}_1^\perp, \mathbf{T}_1^\perp \times \mathbf{N}_1, \mathbf{N}_1]$ for this basis.
4. Now specify \mathbf{N}_2 in the universal frame. Its coordinates in this frame can be given in terms of two angles:

$$\mathbf{N}_2 = (\cos \phi_2 \sin \theta_2, \sin \phi_2 \sin \theta_2, \cos \theta_2). \quad (\text{D.2.35})$$

where θ_2 and ϕ_2 are defined in Figure D.3.

5. We need to define some basis in the plane π_2 that will be the reference for measuring the angle of the epipolar lines therein. We will take as the reference to be the vector $\mathbf{T}_{1,2}^\perp$, which is the projection of \mathbf{T}_1^\perp in the second image, by intersecting π_2 with the special epipolar plane defined by $\overline{\mathbf{e}_1 \mathbf{T}_1}$ and \mathbf{N} (or \mathbf{T}_1^\perp). It is easier to define this special epipolar plane as the $\mathbf{T}_1^\perp \mathbf{N}_1$ plane of the universal coordinate frame. This will be done shortly.
6. Now, given the unit vector \mathbf{v}_1 in the direction of an epipolar line in the first plane, we want to get \mathbf{v}_2 as the intersection of π_2 with the plane $(\mathbf{e}_1 \mathbf{T}_1), \mathbf{v}_1$. Let \mathbf{v}_1 be parametrized by an angle α_1 measured from the \mathbf{T}_1^\perp direction as defined using the right-hand rule. The angle α_2 parametrizing the second pencil of epipolar lines would be defined with respect to our basis of reference $\{\mathbf{T}_{1,2}^\perp, \mathbf{N}_2 \times \mathbf{T}_{1,2}^\perp, \mathbf{N}_2\}$. This way we can write the fundamental relationship between α_1 and α_2 .

Let us carry out the details of the proof.

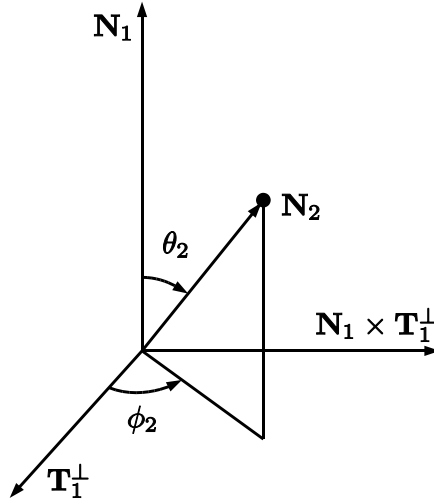


Figure D.3: Spherical coordinates of \mathbf{N}_2 in the universal frame.

Item 5. Given that the second plane π_2 has normal $\mathbf{N}_2(\theta_2, \phi_2)$, we want to intersect it with the plane spanned by \mathbf{T}_1^\perp and \mathbf{N}_1 (therefore having normal $\mathbf{N} \times \mathbf{T}_1^\perp$). The intersection is a line with unit direction $\mathbf{T}_{1,2}^\perp$ which can be written as

$$\mathbf{T}_{1,2}^\perp = \mathbf{N}_2 \times (\mathbf{N}_1 \times \mathbf{T}_1^\perp). \quad (\text{D.2.36})$$

Computing the cross product in universal coordinates, using

$$\mathbf{N}_1 \times \mathbf{T}_1^\perp = (0, 1, 0)^\top, \quad (\text{D.2.37})$$

and using [D.2.35](#), we get:

$$\mathbf{T}_{1,2}^\perp = (\cos \theta_2, 0, -\cos \phi_2 \sin \theta_2)^\top. \quad (\text{D.2.38})$$

Item 6. The vector \mathbf{v}_1 can be written as:

$$\mathbf{v}_1 = (\cos \alpha_1, \sin \alpha_1, 0) \quad (\text{D.2.39})$$

in the universal coordinate frame. Thus, we want a unit vector \mathbf{v}_2 representing the direction of intersection of plane π_2 with the plane spanned by $\mathbf{e}_1 \mathbf{T}_1$ and \mathbf{v}_1 . We can write:

$$\mathbf{v}_2 = \pm \mathbf{N}_2 \times (\mathbf{T}_1 \times \mathbf{v}_1) \quad (\text{D.2.40})$$

where

$$\mathbf{T}_1 = (\sin \theta_1, 0, \cos \theta_1)^\top. \quad (\text{D.2.41})$$

Computing the cross product $\mathbf{T}_1 \times \mathbf{v}_1$ in the universal coordinates, we have that

$$\mathbf{T}_1 \times \mathbf{v}_1 = (-\sin \alpha_1 \cos \theta_1, \cos \alpha_1 \cos \theta_1, \sin \alpha_1 \sin \theta_1)^\top, \quad (\text{D.2.42})$$

and, finally, we can compute the final product:

$$\mathbf{v}_2 = \mathbf{N}_2 \times (\mathbf{T}_1 \times \mathbf{v}_1) \quad (\text{D.2.43})$$

$$\begin{bmatrix} \cos \alpha_1 \cos \theta_1 \cos \theta_2 - \sin \alpha_1 \sin \phi_2 \sin \theta_1 \sin \theta_2 \\ \sin \alpha_1 \cos \phi_2 \sin \theta_1 \sin \theta_2 + \sin \alpha_1 \cos \theta_1 \cos \theta_2 \\ -\sin \alpha_1 \cos \theta_1 \sin \theta_2 \sin \phi_2 - \cos \alpha_1 \cos \theta_1 \sin \theta_2 \cos \phi_2 \end{bmatrix}. \quad (\text{D.2.44})$$

The equation gives the tangent vector \mathbf{v}_2 of the second epipolar line in terms of $\alpha_1, \theta_1, \theta_2, \phi_2$. We can now write the α_2 in terms of a reference coordinate frame in the second image plane. We choose an ordered 3D basis at the second image plane having the vector $\mathbf{T}_{1,2}^\perp$ as the 'x'-axis:

$$\left[\mathbf{T}_{1,2}^\perp, \mathbf{N}_2 \times \mathbf{T}_{1,2}^\perp, \mathbf{N}_2 \right] : \text{ordered basis of } \pi_2. \quad (\text{D.2.45})$$

This way, we can compute both $\sin \alpha_2$ and $\cos \alpha_2$ in terms of α_1 and the essential parameters $\theta_1, \theta_2, \phi_2$ of the system. Assuming that only the slope of the epipolar line is important, but not the direction, we can try to compute $\tan \alpha_2$:

$$\tan \alpha_2 = \frac{\sin \alpha_2}{\cos \alpha_2} = \frac{\mathbf{v}_2 \cdot \mathbf{N}_2 \times \mathbf{T}_{1,2}^\perp}{\mathbf{v}_2 \cdot \mathbf{T}_{1,2}^\perp}. \quad (\text{D.2.46})$$

Let us compute the cosine.

$$\begin{aligned} \cos \alpha_2 &= \mathbf{v}_2 \cdot \mathbf{T}_{1,2}^\perp = \mathbf{v}_2 \cdot (\cos \theta_2, 0, -\cos \phi_2 \sin \theta_2)^\top \\ \cos \alpha_2 &= \cos \alpha_1 \cos \theta_1 \cos^2 \theta_2 - \sin \alpha_1 \sin \theta_1 \sin \theta_2 \cos \theta_2 \sin \phi_2 \\ &\quad + \sin \alpha_1 \cos \theta_1 \sin^2 \theta_2 \sin \phi_2 \cos \phi_2 + \cos \alpha_1 \cos \theta_1 \sin^2 \theta_2 \cos^2 \phi_2. \end{aligned} \quad (\text{D.2.47})$$

Now, the sine first requires the basis vector $\mathbf{N}_2 \times \mathbf{T}_{1,2}^\perp$ to be computed. Using the coordinates of \mathbf{N}_2 and $\mathbf{T}_{1,2}^\perp$ in the universal frame, we can compute their cross product as:

$$\begin{aligned} \mathbf{N}_2 \times \mathbf{T}_{1,2}^\perp &= \pm (\sin^2 \theta_2 \sin \phi_2 \cos \phi_2, -\sin^2 \theta_2 \cos^2 \phi_2 - \cos^2 \theta_2, \sin \theta_2 \cos \theta_2 \sin \phi_2)^\top \\ &= \pm (\sin^2 \theta_2 \sin \phi_2 \cos \phi_2, 1 - \sin^2 \theta_2 \sin^2 \phi_2, \sin \theta_2 \cos \theta_2 \sin \phi_2)^\top, \end{aligned} \quad (\text{D.2.48})$$

in the universal frame. We can now compute the sine

$$\begin{aligned} \sin \alpha_2 &= \mathbf{v}_2 \cdot \mathbf{N}_2 \times \mathbf{T}_{1,2}^\perp \quad (\text{D.2.49}) \\ &= (\cos \alpha_1 \cos \theta_1 \cos \theta_2 - \sin \alpha_1 \sin \theta_1 \sin \theta_2 \sin \phi_2) (\sin^2 \theta_2 \sin \phi_2 \cos \phi_2) \\ &\quad + (\sin \alpha_1 \sin \theta_1 \sin \theta_2 \cos \phi_2 + \sin \alpha_1 \cos \theta_1 \cos \theta_2) (1 - \sin^2 \theta_2 \sin^2 \phi_2) \\ &\quad - (\sin \alpha_1 \cos \theta_1 \sin \theta_2 \sin \phi_2 - \cos \alpha_1 \cos \theta_1 \sin \theta_2 \cos \phi_2) (\sin \theta_2 \cos \theta_2 \sin \phi_2). \end{aligned}$$

Note that we can divide both the $\cos \alpha_2$ and $\sin \alpha_2$ expression above by $\cos \alpha_1$ without affecting their ratio, which is $\tan \alpha_2$. This will enable us to express them in terms of $\tan \alpha_1$. Let us carry this idea out for the cosine:

$$\frac{\cos \alpha_2}{\cos \alpha_1} = \cos \theta_1 \cos^2 \theta_2 - \tan \alpha_1 \sin \theta_1 \sin \theta_2 \cos \theta_2 \sin \phi_2 \quad (\text{D.2.50})$$

$$+ \tan \alpha_1 \cos \theta_1 \sin^2 \theta_2 \sin \phi_2 \cos \phi_2 \quad (\text{D.2.51})$$

$$+ \cos \theta_1 \sin^2 \theta_2 \cos^2 \phi_2 \quad (\text{D.2.52})$$

so we can write this in the form:

$$\frac{\cos \alpha_2}{\cos \alpha_1} := \tilde{c} \tan \alpha_1 + \tilde{d}, \quad (\text{D.2.53})$$

where \tilde{c} and \tilde{d} are constants. We can also write the sine in this form by dividing the equation [D.2.49](#) by $\cos \alpha_1$ and grouping $\tan \alpha_1$ terms, arriving at

$$\frac{\sin \alpha_2}{\cos \alpha_1} := \tilde{a} \tan \alpha_1 + \tilde{b}, \quad (\text{D.2.54})$$

where \tilde{a} and \tilde{b} are constants. Thus

$$\tan \alpha_2 = \frac{\sin \alpha_2}{\cos \alpha_2} = \frac{\tilde{a} \tan \alpha_1 + \tilde{b}}{\tilde{c} \tan \alpha_1 + \tilde{d}}, \quad (\text{D.2.55})$$

which is the epipolar line homography described as a function of four homogeneous parameters $\tilde{a}, \dots, \tilde{d}$, which, when scaled by, say, \tilde{d} , gives 3 coefficients:

$$\tan \alpha_2 = \frac{\sin \alpha_2}{\cos \alpha_2} = \frac{a \tan \alpha_1 + b}{c \tan \alpha_1 + 1}, \quad (\text{D.2.56})$$

with a, b, c functions of the geometric parameters $\theta_1, \theta_2, \phi_2$. ■

Work remains to be done on how to incorporate the intrinsic parameters into a similar geometric analysis.

D.2.3 A Geometric Representation of Epipolar Geometry

As stated in the beginning of this section, we represent the epipolar geometry by two epipoles and a mapping between corresponding epipolar lines. According to Corolary [D.2.4](#), this mapping can be symmetrically represented by an alignment hyperbola:

$$axy + bx + cy + d = 0, \quad (\text{D.2.57})$$

where the epipolar lines are represented by $x = \tan \alpha_1$ in the first image, and $y = \tan \alpha_2$ in the second image. The problem with this equation is that it requires a parametrization of epipolar lines which is singular for vertical lines. In practice, this causes spurious solutions for nearly vertical epipolar lines. One way to fix this is to use an isotropic representation of epipolar lines, namely a unit vector in the direction of the line. Instead of only the tangent of an angle to represent the line, we use the sine and cosine. Substituting $x = \tan \alpha_1 = \sin \alpha_1 / \cos \alpha_1$ and $y = \sin \alpha_2 / \cos \alpha_2$ into [D.2.57](#), and multiplying-out by $\cos \alpha_1 \cos \alpha_2$, we obtain:

$$a \sin \alpha_1 \sin \alpha_1 + b \sin \alpha_1 \cos \alpha_2 + c \cos \alpha_1 \sin \alpha_2 + d \cos \alpha_1 \cos \alpha_2 = 0. \quad (\text{D.2.58})$$

When using tan's, we were parametrizing the epipolar lines by their intersections x and y with a reference line at unit distance from the epipole, namely the tan axis. The new equation using sin and cos corresponds to parametrizing the epipolar lines by their intersections with a unit reference circle around the epipole, namely the trigonometric circle. Conversely, we could have arrived at the above equation directly from this geometric representation, without starting with the equation for tan's.

In a practical implementation, there is use in allowing the reference circles to have arbitrary radius r_1 and r_2 , which can be adjusted in order to guarantee numerical stability in estimating the coefficients of the epipolar alignment. The epipolar lines are then represented by tangent vectors $r_1(\sin \alpha_1, \cos \alpha_1)$ and $r_2(\sin \alpha_2, \cos \alpha_2)$, and the alignment equation becomes:

$$ar_1 \sin \alpha_1 r_2 \sin \alpha_1 + br_1 \sin \alpha_1 r_2 \cos \alpha_2 + cr_1 \cos \alpha_1 r_2 \sin \alpha_2 + dr_1 \cos \alpha_1 r_2 \cos \alpha_2 = 0. \quad (\text{D.2.59})$$

In summary, the advantages of this representation is that it is isotropic and allows more stable estimation by conveniently choosing r_1 and r_2 . The latter reason will be better understood in the next section. Our final geometric representation is stated in the following proposition.

Proposition D.2.6. (*Euclidean representation of epipolar geometry*) *The epipolar geometry can be represented by:*

1. *Two epipoles e_1 and e_2 in cartesian image coordinates.*
2. *Two pencils of epipolar lines as vectors of length r_1 in image 1, and r_2 in image 2, i.e., epipolar lines are represented by their intersections with a circular arc of radius r_1 in image 1 and r_2 in image 2. These vectors are parametrized by angles α_1 and α_2 , respectively, as:*

$$\begin{cases} \mathbf{el}_1(\alpha_1) = r_1(\cos \alpha_1, \sin \alpha_1) = (c_1, s_1) \\ \mathbf{el}_2(\alpha_2) = r_2(\cos \alpha_2, \sin \alpha_2) = (c_2, s_2), \end{cases} \quad (\text{D.2.60})$$

where $c_i := r_i \cos \alpha_i$ and $s_i := r_i \sin \alpha_i$.

3. *The mapping between the epipolar lines as an alignment hyperbola in polar coordinates:*

$$ar_1 \sin \alpha_1 r_2 \sin \alpha_2 + br_1 \sin \alpha_1 r_2 \cos \alpha_2 + cr_1 \cos \alpha_1 r_2 \sin \alpha_2 + dr_1 \cos \alpha_1 r_2 \cos \alpha_2 = 0, \quad (\text{D.2.61})$$

or

$$\begin{bmatrix} c_2 & s_2 \end{bmatrix}^\top \begin{bmatrix} d & b \\ c & a \end{bmatrix} \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} = 0, \quad (\text{D.2.62})$$

where $s_i := r_i \sin \alpha_i$ and $c_i := r_i \cos \alpha_i$, $i = 1, 2$.

D.2.4 Estimating the Epipolar Alignment Curve

The epipolar geometry is determined by seven degrees of freedom: two for each epipole, and three for the alignment of epipolar lines. Three line correspondences are enough to estimate this alignment, given the epipoles. If more lines are available, a least squares solution can be obtained. Note,

however, that this least squares estimation is not guaranteed to be minimizing any sensible cost function in the original domain of epipolar lines (more on this later in this section). More formally, let the epipolar geometry be represented as in Proposition D.2.6, namely the epipoles are represented in cartesian coordinates, and the two pencils of epipolar lines be represented by tangent vectors $\mathbf{e}\mathbf{l}_1(\alpha_1) = r_1(\cos \alpha_1, \sin \alpha_1)$ and $\mathbf{e}\mathbf{l}_2(\alpha_2) = r_2(\cos \alpha_2, \sin \alpha_2)$ normalized to lengths r_1 and r_2 rooted at the epipoles, for images images 1 and 2, respectively. The lengths r_1 and r_2 are independent of the angles α_1 and α_2 . Let us use the short notation $s_i := r_i \sin \alpha_i$ and $c_i := r_i \cos \alpha_i$, $i = 1, 2$. We wish to determine the epipolar alignment given the epipoles and given corresponding epipolar lines. For each corresponding epipolar line $\mathbf{e}\mathbf{l}_1^k, \mathbf{e}\mathbf{l}_2^k$, $k = 1, \dots, m$, we have one linear homogeneous equation on the coefficients (a, b, c, d) of the alignment hyperbola:

$$(s_1^k s_2^k)a + (s_1^k c_2^k)b + (c_1^k s_2^k)c + (c_1^k c_2^k)d = 0. \quad (\text{D.2.63})$$

Notice that the parameters a, b, c, d can be normalized, so we actually have 3DOF to estimate, but we do not normalize them for now. Stacking the m equations, we can write them in matrix form:

$$\begin{bmatrix} s_1^1 s_2^1 & s_1^1 c_2^1 & c_1^1 s_2^1 & c_1^1 c_2^1 \\ s_1^2 s_2^2 & s_1^2 c_2^2 & c_1^2 s_2^2 & c_1^2 c_2^2 \\ s_1^3 s_2^3 & s_1^3 c_2^3 & c_1^3 s_2^3 & c_1^3 c_2^3 \\ \dots & \dots & \dots & \dots \\ s_1^m s_2^m & s_1^m c_2^m & c_1^m s_2^m & c_1^m c_2^m \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (\text{D.2.64})$$

The above system of equations in $\mathbf{h} = (a, b, c, d)$ is traditionally solved by obtaining the null vector of the data matrix A , subject to some normalization constraint on \mathbf{h} , say, $\|\mathbf{h}\| = 1$ [69, p.90]. For perfect measurements, the data matrix has rank 3, and, thus has a 1-dimensional null vector. However, if the corresponding epipolar lines are inaccurate (noisy), then one seeks an approximate solution that minimizes a suitable cost function. What, then, is a suitable cost function? This depends on statistical analysis of maximum likelihood estimation, assuming a statistical model for the noise. However, a simple solution is obtained by minimizing the algebraic norm $\|\mathbf{A}\mathbf{h}\|$ subject to $\|\mathbf{h}\| = 1$, which is equivalent to finding the minimum of the quotient $\|\mathbf{A}\mathbf{h}\|/\|\mathbf{h}\|$. The solution is the (unit) eigenvector of $A^\top A$ with least eigenvalue, or, equivalently, the unit singular vector corresponding to the smallest singular value of A , which is found using SVD. This is actually a basic numerical algorithm that is used in the literature for estimating all kinds of geometric models, even for less trivial problems as the fundamental matrix. The general procedure of fitting a set of homogeneous parameters \mathbf{h} minimizing an algebraic residual through SVD is called the Direct Linear Transform algorithm (DLT) in the literature.

It is well-known that the DLT algorithm for fitting models to *point* correspondences can be very unstable and coordinate-dependent, and, thus, a normalization of the coordinates has to be performed prior to solving the linear system in the case of homography estimation and fundamental matrix estimation [69, 71]. The normalization consists in translating data points such that their mean is 0 (or centroid is at origin), and the average point has unit coordinates in all directions. The

resulting algorithm is called the normalized DLT. In [71], Hartley had proved that this transformation improves the conditioning of the data matrix A (also called the *design matrix*). More recent literature claims that normalization only alleviates the problem of stability, but does not get rid of it [66,146]. It is standard practice to use the linear solution as initial input to an iterative optimization procedure that minimizes geometric cost. The geometric cost is the cost giving the maximum likelihood estimate under assumptions of independent Gaussian noise of equal variance in the data. It is called geometric since it has meaningful interpretation as Euclidean distances.

In our case, we are fitting a hyperbola to correspondences between epipolar lines. We had hoped that our geometric representation naturally alleviates this stability problem of the linear algorithm. However, experiments showed that this is not the case. In these experiments, we were given the epipoles together with 8 corresponding (noisy) points, and the task was to estimate the best alignment between epipolar lines. The first attempt was to write 8 equations as in D.2.64, and solve by minimizing $\|A\mathbf{h}\|$ subject to $\|\mathbf{h}\| = 1$ using SVD. However, for some point configurations, a small perturbation in the data caused a completely useless epipolar alignment. Our final solution was, then, to select all subsets of 3 epipolar lines, get the solution as the null vector from the data matrix A for each of these subsets, and measure the geometric error to all the remaining 5 corresponding points. The final estimate was, then, the one minimizing geometric error. This really improved the stability of the problem, and in our Monte Carlo experiments we always obtained reasonable fits. However, this was time consuming: one has to select all subsets of 3 correspondences out of 8, solve a linear system for each of them, and measure geometric cost to the remaining correspondences. Our current alternative method, then, is to use the most widely separated epipolar line correspondences to estimate the epipolar alignment. This also has given reasonably stable estimates, although not quite as good as the exhaustive search of all subsets.

A note on the cost function being minimized. Each data point for the least squares estimation is an epipolar line correspondence. As far as the numerical estimation is concerned, these data points are nothing but x and y coordinates in a Cartesian system representing samples of the alignment hyperbola. Even though the Euclidean distance to the alignment hyperbola might be minimized by hyperbola fitting, this distance does not necessarily reflect a meaningful (i.e. maximum likelihood) distance between predicted and measured epipolar lines.

The four epipolar line consistency rule

As already mentioned, three line correspondences are enough to estimate the epipolar homography, given the epipoles. Any extra line can be used for verification.

Proposition D.2.7. (The four epipolar line consistency rule): *Consider two images with epipoles \mathbf{e}_1 and \mathbf{e}_2 , respectively. Let the epipolar lines be represented by angles α_i and β_i around epipoles 1 and 2, respectively, their tangents being $x = \tan \alpha_i$ and $y = \tan \beta_i$. Then, for any four corresponding*

epipolar lines, we have

$$\begin{vmatrix} x_1y_1 & x_1 & y_1 & 1 \\ x_2y_2 & x_2 & y_2 & 1 \\ x_3y_3 & x_3 & y_3 & 1 \\ x_4y_4 & x_4 & y_4 & 1 \end{vmatrix} = 0. \quad (\text{D.2.65})$$

That is, the determinant is zero if and only if there is some epipolar geometry consistent with the four lines.

Proof. From equation (D.2.28), we have:

$$ax_iy_i + bx_i + cy_i + d = 0, \quad (\text{D.2.66})$$

for $i = 1, \dots, 4$. We can write these equations as:

$$\begin{bmatrix} x_1y_1 & x_1 & y_1 \\ x_2y_2 & x_2 & y_2 \\ x_3y_3 & x_3 & y_3 \\ x_4y_4 & x_4 & y_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} d \\ d \\ d \\ d \end{bmatrix} \quad (\text{D.2.67})$$

or

$$\begin{bmatrix} x_1y_1 & x_1 & y_1 & 1 \\ x_2y_2 & x_2 & y_2 & 1 \\ x_3y_3 & x_3 & y_3 & 1 \\ x_4y_4 & x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = 0 \quad (\text{D.2.68})$$

The above system has non-zero solutions if and only if the 4×4 matrix is rank-deficient, *i.e.*, its determinant is zero. ■

The four epipolar line consistency rule can be used in an algorithm that estimates epipolar geometry from point correspondences. Given seven corresponding points and a putative position for the epipoles, one has 7 corresponding epipolar lines – one for each point correspondence. Every four lines provide a constraint, and there are four independent constraints given 7 points. Four constraints of the kind shown proposition D.2.7 can in principle be used to find the epipoles \mathbf{e}_1 and \mathbf{e}_2 .

It turns out that the four-line consistency equation expresses nothing more than the invariance of the cross-ratio under the epipolar transformation.

Corollary D.2.8. *The four-line consistency rule means that the cross ratio of four epipolar lines must be preserved by the epipolar alignment.*

The importance of cross ratios in the literature is vast; by studying its properties we can arrive at better estimation of the epipolar geometry.

D.2.5 Direct Approach to Solving the Geometric Equations

In this section, we attempt to solve the geometric epipolar equations directly, given corresponding points. By using the geometric representation where the epipoles and epipolar homography are explicit, we expect to develop a better approach for epipolar geometry estimation.

Since we are working with two views, the notation can be simplified. Let $\mathbf{e} = (x, y)$ be the epipole in image 1 and $\bar{\mathbf{e}}(\bar{x}, \bar{y})$ the epipole in image 2. Similarly, let $\mathbf{p} = (\xi, \eta)$ and $\bar{\mathbf{p}} = (\bar{\xi}, \bar{\eta})$ be points in images 1 and 2, respectively. The equations of Proposition D.2.6 can be written as:

$$(\bar{\mathbf{p}} - \bar{\mathbf{e}})^\top H(\mathbf{p} - \mathbf{e}) = 0, \quad (\text{D.2.69})$$

where

$$H = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \quad (\text{D.2.70})$$

We wish to solve this equation using 7 point correspondences. One approach to solving this equation directly is to first use 2 point correspondences to solve for \mathbf{e} , then use 4 more to solve for H and, finally, using 2 correspondences for solving a system of 2 polynomials in $\bar{\mathbf{e}}$. This approach has the advantage that only linear equations have to be solved up to the last two. From [103], we know the last 2 equations must be two cubics.

Let us carry out this approach. First, rewrite the above equation as

$$(\bar{\mathbf{p}} - \bar{\mathbf{e}})^\top H\mathbf{e} = (\bar{\mathbf{p}} - \bar{\mathbf{e}})^\top H\mathbf{p}, \quad (\text{D.2.71})$$

and we try to solve for the 2D vector $H\mathbf{e}$ in terms of the other unknowns H and $\bar{\mathbf{e}}$ and plug back into the equation. In other words, we just need to eliminate $H\mathbf{e}$ using two point correspondences:

$$\begin{cases} (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top H\mathbf{e} = (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top H\mathbf{p}_1 \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top H\mathbf{e} = (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top H\mathbf{p}_2, \end{cases} \quad (\text{D.2.72})$$

We can now solve for $H\mathbf{e}$ using Cramer's rule and substitute back. However, since we will have to multiply both sides by the denominator given by Cramer's rule anyways, we can obtain the resulting equation more directly if we introduce an auxiliary variable w to turn the above inhomogeneous system in $H\mathbf{e}$ into a homogeneous one in $(wH\mathbf{e}, w)^\top$, and then solving for for the null vector using a simple cross-product operation. Before carrying this out, let us define the Kronecker product of two vectors as:

$$\mathbf{v} \otimes \mathbf{w} := (w_1v_1, w_1v_2, w_2v_1, w_2v_2)^\top. \quad (\text{D.2.73})$$

Using this operator, we can write

$$\mathbf{v}^\top H\mathbf{w} = (\mathbf{v} \otimes \mathbf{w})^\top \mathbf{h}, \quad (\text{D.2.74})$$

where $\mathbf{h} = (a, b, c, d)^\top$ is the matrix H rewritten as a vector. This way, equation D.2.71 can be written as:

$$(\bar{\mathbf{p}} - \bar{\mathbf{e}})^\top H\mathbf{e} = [\mathbf{p} \otimes (\bar{\mathbf{p}} - \bar{\mathbf{e}})]^\top \mathbf{h}, \quad (\text{D.2.75})$$

and the system of equations given two points can be written as

$$\begin{cases} (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top H\mathbf{e} = [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \mathbf{h} \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top H\mathbf{e} = [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \mathbf{h} \end{cases} \quad (\text{D.2.76})$$

and we try to solve for $H\mathbf{e}$ (and eliminate it). We can turn this system into a homogeneous form by multiplying both sides of the equations by an auxiliary variable w :

$$\begin{cases} (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top wH\mathbf{e} = [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \mathbf{h} w \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top wH\mathbf{e} = [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \mathbf{h} w \end{cases} \quad (\text{D.2.77})$$

So that we can write this as

$$\begin{bmatrix} (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top & [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \mathbf{h} \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top & [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \mathbf{h} \end{bmatrix} \begin{bmatrix} wH\mathbf{e} \\ w \end{bmatrix} = 0 \quad (\text{D.2.78})$$

This enables us to write the solution as a cross-product, and we conveniently obtain w (which is arbitrary) such that the equations have no denominator. The solution obtained by conveniently setting w can be written as:

$$\begin{bmatrix} wH\mathbf{e} \\ w \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{p}}_1 - \bar{\mathbf{e}} \\ [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \mathbf{h} \end{bmatrix} \times \begin{bmatrix} \bar{\mathbf{p}}_2 - \bar{\mathbf{e}} \\ [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \mathbf{h} \end{bmatrix}. \quad (\text{D.2.79})$$

Using determinant notation, this is equivalent to

$$\begin{bmatrix} wH\mathbf{e} \\ w \end{bmatrix} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top & [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \mathbf{h} \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top & [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \mathbf{h} \end{vmatrix} \quad (\text{D.2.80})$$

Therefore, when this is substituted back into [D.2.75](#) (after multiplying both sides by w), we obtain:

$$\left[(\bar{\mathbf{p}} - \bar{\mathbf{e}})^\top \mid [\mathbf{p} \otimes (\bar{\mathbf{p}} - \bar{\mathbf{e}})]^\top \mathbf{h} \right]^\top \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top & [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \mathbf{h} \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top & [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \mathbf{h} \end{vmatrix} = 0 \quad (\text{D.2.81})$$

This is just the scalar triple product, and can be written as a single determinant:

$$\begin{vmatrix} (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top & [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \mathbf{h} \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top & [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \mathbf{h} \\ (\bar{\mathbf{p}}_i - \bar{\mathbf{e}})^\top & [\mathbf{p}_i \otimes (\bar{\mathbf{p}}_i - \bar{\mathbf{e}})]^\top \mathbf{h} \end{vmatrix} = 0 \quad (\text{D.2.82})$$

This equation involving a 3×3 determinant we call the *three point constraint*.

We now need to eliminate \mathbf{h} from equation [D.2.82](#) using three equations instantiated for $i = 3, 4, 5$. First, let us rewrite such equation as a linear system form in \mathbf{h} . This can be done by expanding the determinant by the third column, and can be shown to be:

$$\left[\begin{vmatrix} (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top \\ (\bar{\mathbf{p}}_i - \bar{\mathbf{e}})^\top \end{vmatrix}, - \begin{vmatrix} (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top \\ (\bar{\mathbf{p}}_i - \bar{\mathbf{e}})^\top \end{vmatrix}, \begin{vmatrix} (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top \\ (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top \end{vmatrix} \right] \begin{bmatrix} [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})]^\top \\ [\mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top \\ [\mathbf{p}_i \otimes (\bar{\mathbf{p}}_i - \bar{\mathbf{e}})]^\top \end{bmatrix} \mathbf{h} = 0 \quad (\text{D.2.83})$$

We can obtain an equivalent equation as follows. Let the operator $\text{rev}(\mathbf{v}) = (\mathbf{v}_2, \mathbf{v}_1)^\top$, *i.e.*, the operator which reverses the coordinates. From the expression of the cross product using determinants, we can obtain expressions for w and wHe , which we can plug back into original equation to obtain:

$$\begin{aligned} & \{(\bar{\mathbf{p}}_i - \bar{\mathbf{e}})^\top [-\text{rev}(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}}), \text{rev}(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})] \cdot [\mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}}), \mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})]^\top - \\ & \det[\bar{p}_1 - \bar{e}, \bar{p}_2 - \bar{e}] [\mathbf{p} \otimes (\bar{\mathbf{p}} - \bar{\mathbf{e}})]^\top\} \mathbf{h} = 0. \end{aligned} \quad (\text{D.2.84})$$

The vector inside curly brackets $\{\}$ has 4 entries, each one being a cubic polynomial in \bar{x} and \bar{y} . We can write it explicitly as:

Coordinate 1:

$$\begin{aligned} & \bar{x}^2 \bar{y} (2\xi_1) + \\ & \bar{x} \bar{y} (-2\xi_1 \bar{\xi}_1 - \xi_1 \bar{\xi}_i - \xi_1 \bar{\xi}_2 - \xi_2 \bar{\xi}_1 + \xi_2 \bar{\xi}_i - \xi_i \bar{\xi}_1 + \xi_i \bar{\xi}_2) + \\ & \bar{x}^2 (\xi_1 \bar{\eta}_i - \xi_1 \bar{\eta}_2 + \xi_2 \bar{\eta}_i - \xi_2 \bar{\eta}_1 - \xi_i \bar{\eta}_2 + \xi_i \bar{\eta}_1) + \\ & \bar{x} (\xi_1 \bar{\xi}_1 \bar{\eta}_i + \xi_1 \bar{\xi}_1 \bar{\eta}_2 - \xi_1 \bar{\xi}_i \bar{\eta}_2 + \xi_1 \bar{\eta}_i \bar{\xi}_2 + \xi_2 \bar{\xi}_2 \bar{\eta}_i - \xi_2 \bar{\xi}_2 \bar{\eta}_1 - \xi_2 \bar{\xi}_i \bar{\eta}_1 + \xi_2 \bar{\eta}_i \bar{\xi}_1 - \xi_i \bar{\xi}_2 \bar{\eta}_1 + \xi_i \bar{\xi}_1 \bar{\eta}_2 - \xi_i \bar{\xi}_i \bar{\eta}_1 + \xi_i \bar{\xi}_i \bar{\eta}_2) + \\ & \bar{y} (\xi_1 \bar{\xi}_1 \bar{\xi}_i - \xi_1 \bar{\xi}_1 \bar{\xi}_2 + \xi_2 \bar{\xi}_2 \bar{\xi}_1 - \xi_2 \bar{\xi}_2 \bar{\xi}_i - \xi_i \bar{\xi}_i \bar{\xi}_2 + \xi_i \bar{\xi}_i \bar{\xi}_1) + \\ & 1 (\xi_1 \bar{\xi}_1 \bar{\xi}_i \bar{\eta}_2 - \xi_1 \bar{\xi}_1 \bar{\eta}_i \bar{\xi}_2 + \xi_2 \bar{\xi}_2 \bar{\xi}_i \bar{\eta}_1 - \xi_2 \bar{\xi}_2 \bar{\eta}_i \bar{\xi}_1 - \xi_i \bar{\xi}_i \bar{\xi}_1 \bar{\eta}_2 + \xi_i \bar{\xi}_i \bar{\xi}_2 \bar{\eta}_1) \end{aligned} \quad (\text{D.2.85})$$

Coordinate 2:

$$\begin{aligned} & \bar{x}^2 \bar{y} (2\eta_1) + \\ & \bar{x} \bar{y} (-2\eta_1 \bar{\xi}_1 - \xi_1 \bar{\eta}_i - \eta_1 \bar{\xi}_2 - \eta_2 \bar{\xi}_1 + \eta_2 \bar{\xi}_i - \eta_i \bar{\eta}_1 + \eta_i \bar{\eta}_2) + \\ & \bar{x}^2 (\eta_1 \bar{\eta}_i - \eta_1 \bar{\eta}_2 + \eta_2 \bar{\eta}_i - \eta_2 \bar{\eta}_1 - \eta_i \bar{\eta}_2 + \eta_i \bar{\eta}_1) + \\ & \bar{x} (\eta_1 \bar{\xi}_1 \bar{\eta}_i + \eta_1 \bar{\xi}_1 \bar{\eta}_2 - \eta_1 \bar{\xi}_i \bar{\eta}_2 + \eta_1 \bar{\eta}_i \bar{\xi}_2 + \eta_2 \bar{\xi}_2 \bar{\eta}_i - \eta_2 \bar{\xi}_2 \bar{\eta}_1 - \eta_2 \bar{\xi}_i \bar{\eta}_1 + \eta_2 \bar{\eta}_1 \bar{\xi}_2 - \eta_i \bar{\xi}_2 \bar{\eta}_1 + \eta_i \bar{\xi}_1 \bar{\eta}_2 - \eta_i \bar{\xi}_i \bar{\eta}_1 + \xi_i \bar{\xi}_i \bar{\eta}_2) + \\ & \bar{y} (\eta_1 \bar{\xi}_1 \bar{\xi}_i - \eta_1 \bar{\xi}_1 \bar{\xi}_2 + \eta_2 \bar{\xi}_2 \bar{\xi}_1 - \eta_2 \bar{\xi}_2 \bar{\xi}_i - \eta_i \bar{\xi}_i \bar{\xi}_2 + \eta_i \bar{\xi}_i \bar{\xi}_1) + \\ & 1 (\eta_1 \bar{\xi}_1 \bar{\xi}_i \bar{\eta}_2 - \eta_1 \bar{\xi}_1 \bar{\eta}_i \bar{\xi}_2 + \eta_2 \bar{\xi}_2 \bar{\xi}_i \bar{\eta}_1 - \eta_2 \bar{\xi}_2 \bar{\eta}_i \bar{\xi}_1 - \eta_i \bar{\xi}_i \bar{\xi}_1 \bar{\eta}_2 + \eta_i \bar{\xi}_i \bar{\xi}_2 \bar{\eta}_1) \end{aligned} \quad (\text{D.2.86})$$

Coordinate 3:

$$\begin{aligned} & \bar{x} \bar{y}^2 (2\xi_1) + \\ & \bar{x} \bar{y} (-2\xi_1 \bar{\eta}_1 - \xi_1 \bar{\eta}_i - \xi_1 \bar{\eta}_2 - \xi_2 \bar{\eta}_i + \xi_2 \bar{\eta}_1 - \xi_i \bar{\eta}_2 + \xi_i \bar{\eta}_1) + \\ & \bar{y}^2 (-\xi_1 \bar{\xi}_i - \xi_1 \bar{\xi}_2 - \xi_2 \bar{\xi}_1 + \xi_2 \bar{\xi}_i - \xi_i \bar{\xi}_1 + \xi_i \bar{\xi}_2) + \\ & \bar{x} (\xi_1 \bar{\eta}_1 \bar{\eta}_i + \xi_1 \bar{\eta}_1 \bar{\eta}_2 + \xi_2 \bar{\eta}_2 \bar{\eta}_i - \xi_2 \bar{\eta}_2 \bar{\eta}_1 - \xi_i \bar{\eta}_i \bar{\eta}_1 + \xi_i \bar{\eta}_i \bar{\eta}_2) + \\ & \bar{y} (\xi_1 \bar{\eta}_1 \bar{\xi}_1 - \xi_1 \bar{\eta}_1 \bar{\xi}_2 - \xi_1 \bar{\xi}_i \bar{\eta}_2 + \xi_1 \bar{\eta}_i \bar{\xi}_2 + \xi_2 \bar{\eta}_2 \bar{\xi}_1 - \xi_2 \bar{\eta}_2 \bar{\xi}_i - \xi_2 \bar{\xi}_i \bar{\eta}_1 + \xi_2 \bar{\eta}_i \bar{\xi}_1 - \xi_i \bar{\xi}_2 \bar{\eta}_1 + \xi_i \bar{\xi}_1 \bar{\eta}_2 - \xi_i \bar{\eta}_i \bar{\xi}_2 + \xi_i \bar{\eta}_i \bar{\xi}_1) + \\ & 1 (\xi_1 \bar{\eta}_1 \bar{\xi}_i \bar{\eta}_2 - \xi_1 \bar{\eta}_1 \bar{\eta}_i \bar{\xi}_2 + \xi_2 \bar{\eta}_2 \bar{\xi}_1 \bar{\eta}_i - \xi_2 \bar{\eta}_2 \bar{\eta}_i \bar{\xi}_1 - \xi_i \bar{\eta}_i \bar{\xi}_1 \bar{\eta}_2 + \xi_i \bar{\eta}_i \bar{\xi}_2 \bar{\eta}_1) \end{aligned} \quad (\text{D.2.87})$$

Coordinate 4:

$$\begin{aligned}
& \bar{x}\bar{y}^2(2\eta_1)+ \\
& \bar{x}\bar{y}(-2\eta_1\bar{\eta}_1 - \eta_1\bar{\eta}_i - \eta_1\bar{\eta}_2 - \eta_2\bar{\eta}_i + \eta_2\bar{\eta}_1 - \eta_i\bar{\eta}_2 + \eta_i\bar{\eta}_1)+ \\
& \bar{y}^2(-\eta_1\bar{\xi}_i - \eta_1\bar{\xi}_2 - \eta_2\bar{\xi}_1 + \eta_2\bar{\xi}_i - \eta_i\bar{\xi}_1 + \eta_i\bar{\xi}_2)+ \\
& \bar{x}(\eta_1\bar{\eta}_1\bar{\eta}_i + \eta_1\bar{\eta}_1\bar{\eta}_2 + \eta_2\bar{\eta}_2\bar{\eta}_i - \eta_2\bar{\eta}_2\bar{\eta}_1 - \eta_i\bar{\eta}_i\bar{\eta}_1 + \eta_i\bar{\eta}_i\bar{\eta}_2)+ \\
& \bar{y}(\eta_1\bar{\eta}_1\bar{\xi}_1 - \eta_1\bar{\eta}_1\bar{\xi}_2 - \eta_1\bar{\xi}_i\bar{\eta}_2 + \eta_1\bar{\eta}_i\bar{\xi}_2 + \eta_2\bar{\eta}_2\bar{\xi}_1 - \eta_2\bar{\eta}_2\bar{\xi}_i - \eta_2\bar{\xi}_i\bar{\eta}_1 + \eta_2\bar{\eta}_i\bar{\xi}_1 - \eta_i\bar{\xi}_2\bar{\eta}_1 + \eta_i\bar{\xi}_1\bar{\eta}_2 - \eta_i\bar{\eta}_i\bar{\xi}_2 + \eta_i\bar{\eta}_i\bar{\xi}_1)+ \\
& 1(\eta_1\bar{\eta}_1\bar{\xi}_i\bar{\eta}_2 - \eta_1\bar{\eta}_1\bar{\eta}_i\bar{\xi}_2 + \eta_2\bar{\eta}_2\bar{\xi}_i\bar{\eta}_1 - \eta_2\bar{\eta}_2\bar{\eta}_i\bar{\xi}_1 - \eta_i\bar{\eta}_i\bar{\xi}_1\bar{\eta}_2 + \eta_i\bar{\eta}_i\bar{\xi}_2\bar{\eta}_1)
\end{aligned} \tag{D.2.88}$$

$$\begin{bmatrix} \bar{x}^2\bar{y} & \bar{x}\bar{y}^2 & \bar{x}^2 & \bar{x}\bar{y} & \bar{y}^2 & \bar{x} & \bar{y} & 1 \end{bmatrix} \begin{bmatrix} \text{col 1} & \text{col 2} & \text{col 3} & \text{col 4} \end{bmatrix} \mathbf{h} = 0. \tag{D.2.89}$$

We must use 3 such equations in order to get rid of \mathbf{h} . We will have 3 equations of this form instantiated for $i = 3, 4, 5$, which we can solve using wedge product determinant. Plugging back into the general equation for arbitrary i , we obtain a 4×4 determinant equal to zero equation. Each of the entries in the determinant is a cubic, so, in principle, its the expression will have degree 81. This, somehow should be equivalent to a cubic if the classic literature is correct.

Let us explicitly write the final system of 2 polynomials in 2 variables $\bar{e} = \bar{x}, \bar{y}$ using determinant notation:

$$\begin{vmatrix} \left[\begin{array}{c} |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_3 - \bar{\mathbf{e}})^\top| \end{array} \right], - \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_3 - \bar{\mathbf{e}})^\top| \end{array} \right], \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \end{array} \right] \left[\begin{array}{ccc} \mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}}) & \mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}}) & \mathbf{p}_3 \otimes (\bar{\mathbf{p}}_3 - \bar{\mathbf{e}}) \end{array} \right]^\top \\ \left[\begin{array}{c} |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_4 - \bar{\mathbf{e}})^\top| \end{array} \right], - \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_4 - \bar{\mathbf{e}})^\top| \end{array} \right], \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \end{array} \right] \left[\begin{array}{ccc} \mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}}) & \mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}}) & \mathbf{p}_4 \otimes (\bar{\mathbf{p}}_4 - \bar{\mathbf{e}}) \end{array} \right]^\top \\ \left[\begin{array}{c} |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_5 - \bar{\mathbf{e}})^\top| \end{array} \right], - \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_5 - \bar{\mathbf{e}})^\top| \end{array} \right], \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \end{array} \right] \left[\begin{array}{ccc} \mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}}) & \mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}}) & \mathbf{p}_5 \otimes (\bar{\mathbf{p}}_5 - \bar{\mathbf{e}}) \end{array} \right]^\top \\ \left[\begin{array}{c} |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_i - \bar{\mathbf{e}})^\top| \end{array} \right], - \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_i - \bar{\mathbf{e}})^\top| \end{array} \right], \left[\begin{array}{c} |(\bar{\mathbf{p}}_1 - \bar{\mathbf{e}})^\top| \\ |(\bar{\mathbf{p}}_2 - \bar{\mathbf{e}})^\top| \end{array} \right] \left[\begin{array}{ccc} \mathbf{p}_1 \otimes (\bar{\mathbf{p}}_1 - \bar{\mathbf{e}}) & \mathbf{p}_2 \otimes (\bar{\mathbf{p}}_2 - \bar{\mathbf{e}}) & \mathbf{p}_i \otimes (\bar{\mathbf{p}}_i - \bar{\mathbf{e}}) \end{array} \right]^\top \end{vmatrix} = 0, \tag{D.2.90}$$

and our system is written as 2 such equations instantiated to $i = 6, 7$. How can we reduce (D.2.90) as a cubic in \bar{e} ? This is left for future work.

D.3 Qualitative Aspects and Epipolar Ordering

In this section, we propose and explore a new method for computing epipolar geometry that addresses the issues of traditional approaches listed in the introduction.

D.3.1 Partitioning via Ordering Invariance

The central idea of this section is to qualitatively constrain the epipolar geometry based on the notion of order. Given point correspondences between two images, if the epipolar lines in image 1 are

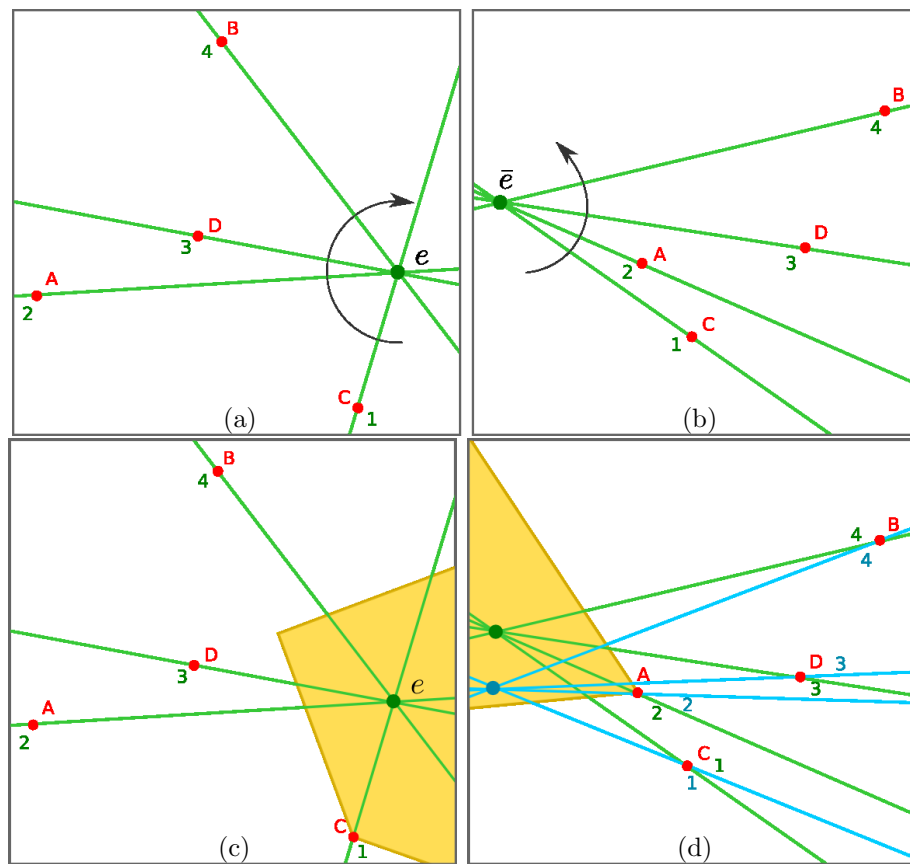


Figure D.4: Ordering invariant.

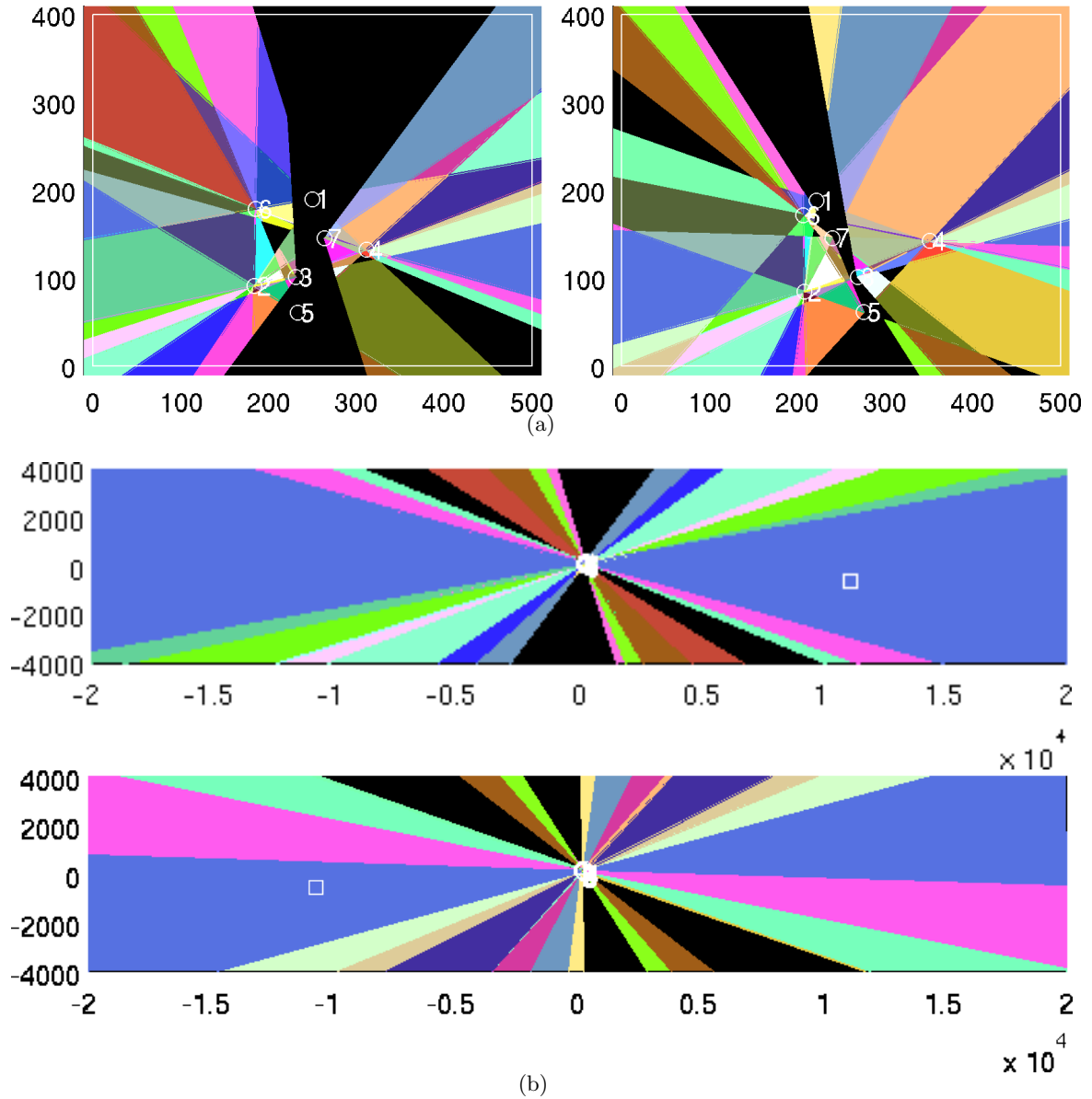


Figure D.5: Polygon partitioning of the space of putative epipoles: (a) restricted to the two image domains, and (b) restricted to a more extended region of the 2D plane with the true epipole shown as a small square. Polygons are color-coded to represent distinct equivalence classes of putative epipoles. Black regions represent impossible solutions violating ordering. Notice how the true epipole satisfies the ordering represented by the blue polygon.

ordered in clockwise manner, then the corresponding epipolar lines in image 2 must also be ordered either clockwise or counterclockwise. Figure D.4(a,b) illustrates this: suppose we order the epipolar lines through points A, \dots, D in image 1 in a clockwise fashion; then the corresponding points in image 2 must also have their respective epipolar lines ordered either clockwise or counterclockwise. This is related to the following theorem.

Theorem D.3.1. *Given point correspondences in two views, then corresponding epipolar lines partition each image points into two sets which are corresponding.*

Proof. Corresponding epipolar lines define a 3D epipolar plane. Features in 2D correspond to points in 3D, which are either above or below the given epipolar plane. Similarly, in each image the points are partitioned in two sets, each on each side of the epipolar lines, and the partitioning is the same for both images. Since we only used intersection of points in this proof, the theorem is also valid in the case of uncalibrated cameras looking at the same scene. ■

We can use this constraint to rule out impossible solutions to epipolar geometry. For a putative epipole e in the plane of image 1, there might be no epipoles \bar{e} in image 2 that preserve the same ordering of epipolar lines. In this case we can discard any epipolar geometry having the first epipole at e .

If there happens to be a set of epipoles \bar{e} in the plane of image 2 that can preserve the same ordering of epipolar lines as e for the given point correspondences, then this set can be shown to be a polygonal region, as illustrated in Figure D.4(c).

There are other epipoles in image 1 that can produce the same clockwise ordering of epipolar lines as e , and they also form a polygon in image 1. Therefore we have two polygons, one for each image, of corresponding epipoles that preserve the same ordering, as shown in Figure D.4(d,e). These corresponding polygons are the equivalence class of the epipoles e, \bar{e} , under the equivalence relation “induces the same ordering” for the given points. We thus rule out any epipolar geometry having an epipole in one of these polygons but not in the other polygon.

A geometric algorithm can be used to compute the partition of the space of epipoles induced by such an equivalence relation. Figure D.5 shows one such partition using 7 point correspondences across two views separated by 30 degrees: (a) Shows two views of 7 points (b) shows the zoomed-out version of the same views with the ground-truth epipole as a white square. Polygons where corresponding epipoles can be are shown in random colors. Corresponding polygons have the same colors. No epipole is allowed to be in the black regions.

The ordering partition greatly reduces the search for epipoles by ruling out major parts of the 4D space of epipoles that violate ordering, the remaining possibilities being partitioned into non-overlapping polytopes. In other words, we reduced the space of epipoles from 4D combinations to a finite number of smaller 4D problems for each polygon in correspondence. This is explored in conjunction with an idea described next in order to further nail down the epipolar geometry.

D.3.2 Cost Function on Epipoles

Given a set of 4 or more point correspondences, solving epipolar geometry can be formulated as the search for the optimal position of the epipoles. The optimal solution is the one minimizing a geometric cost function, such as the sum of distances of point correspondences to the respective epipolar lines. Such a cost function is known to have quite intricate patterns, so that iterative optimization gets trapped in local minima, even when initialized by the 7 or 8 point algorithms [40]. The qualitative partitioning of the space of epipoles greatly reduces the 4D search for epipoles, generating a number of smaller constrained minimization problems for each of the feasible polygons, as described next.

D.3.3 Hierarchical Search

Given a polygonal partition of the space of epipoles into equivalence classes, we further rule-out polygons that are sure not to contain the minimum. This is done by inferring properties of the cost function on the whole polygon, based on the cost of the centroid solution. If the centroid cost is so high that, for the given size of polygon, all other epipole solutions within the polygon are also bad, then the whole polygon is discarded. For the remaining polygons, either subdivide and repeat the process for each polygon, or perform a local optimization if conditions for this are met (such as a small enough partition).

Currently, the way we perform the local optimization is in an experimental stage. Given epipoles and three points, we can obtain the epipolar homography using a linear method D.2.4. If we are working with 7 points, then the geometric cost to the remaining set of points can be used to evaluate how good a given epipole position is. This can be used to rank-order the polygons through the geometric cost of placing epipoles at their centroids. For the top 5 polygons, we perform non-linear optimization of epipole positions minimizing geometric cost using Levenberg-Marquadt. We take as the final epipole solution the one having smallest geometric cost after this optimization. In order to optimize within finite polygonal regions, and in order to sensibly compute centroids, we have explored the use of a spherical representation for each image plane, using either stereographic projection or gnomonic projections with the center of projection being the camera focal point.

The key to the success of the algorithm resides in the stability of computing the polygonal partition as demonstrated in the following experiment. We compute the epipolar geometry solely based on rank-ordering rough solutions consisting of epipoles placed at the centroids of corresponding polygons, running a non-linear least squares optimization on the top 5, and outputting the solutions with smallest cost. This is not meant to be a practical method, its purpose is just to illustrate the stability of the polygons. Figure D.6 shows the best geometric error of the polygonal partition-based method and the solution from the 8 point algorithm.

D.3.4 Contributions of this Section

The potential contributions and advantages of the ideas presented in this section are:

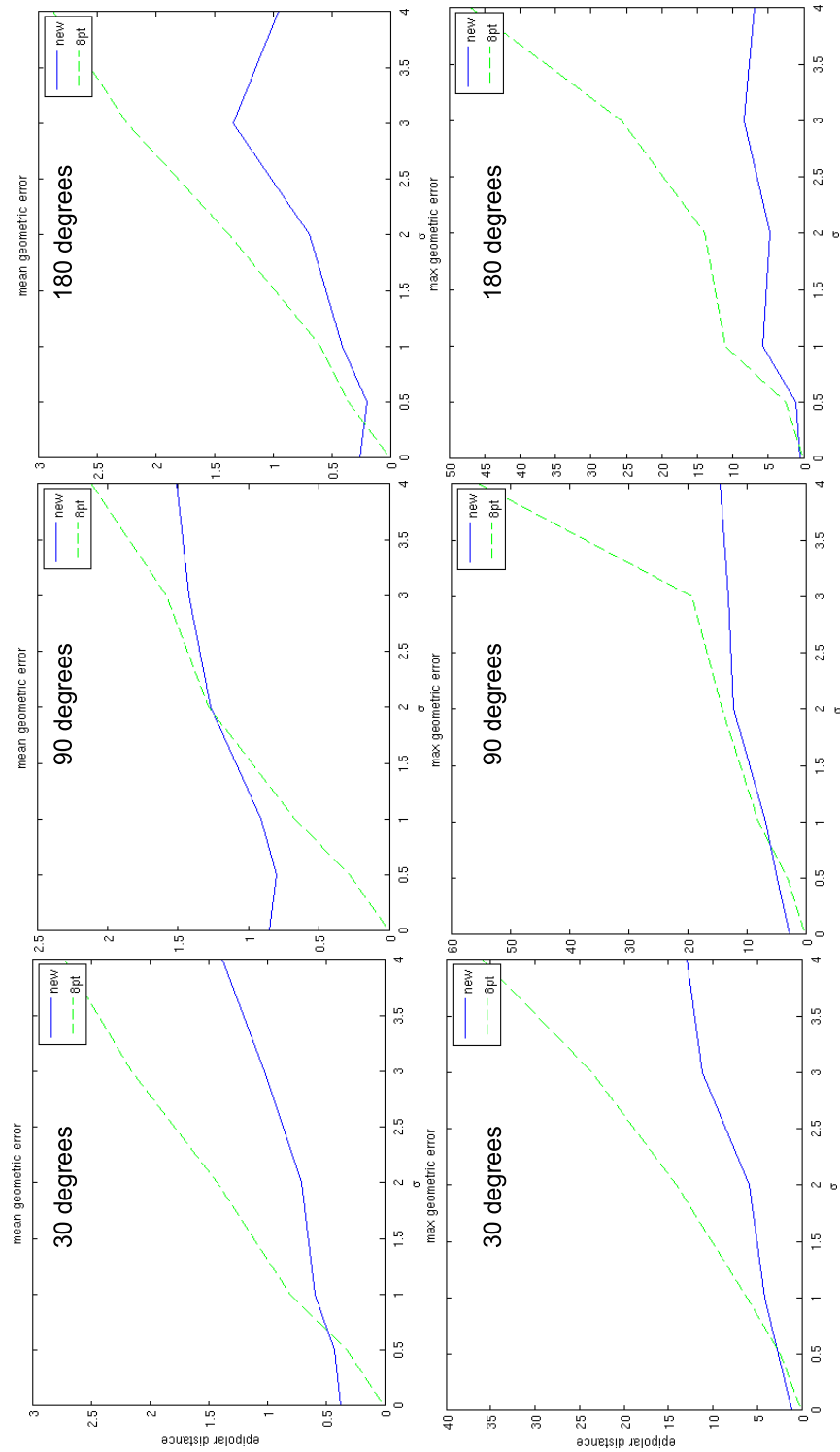


Figure D.6: Stability plots. Even a simple solution of epipolar geometry using just the centroids of our polygons clearly outperforms the 8 point algorithm for medium to large localization error levels.

- Computation of epipolar geometry that is more stable than previous methods for high levels of localization noise.
- Our polygonal partition can be used to prune down the 4D space of epipoles given at least three points.
- Our method can be used to constrain optimization of epipoles to a bounded region that is computable by a stable algorithm.
- Our method can be used in a RANSAC strategy to prune correspondences with less than the minimum of 7 points (3 or more points). Starting from a RANSAC search for 3 reliable correspondences (only 35 runs), our approach allows one to iteratively add another reliable correspondence until 7 or 8 are reached, or to discard the 3 correspondences if it is not possible to add a fourth one. This gives increased speed, but without having to search for 7 individually. It also gives increased robustness: could even add more than 7 points, up to *e.g.* 15 points so that the bumps in the cost function are smoothed out.
- Robustness to coordinate changes.
- Directly-interpretable geometric entities, leading to *e.g.* coordinate invariance.
- The approach might be a starting point for bringing geometric insight and efficient geometric algorithms to problems in 3D computer vision and geometric fitting. Many problems have structure similar to epipolar geometry estimation [82,90]. Insights from computational geometry and computer graphics have already inspired optimal algorithms in computer vision, *e.g.* visual hull construction, optimal exact Euclidean distance transforms based on Voronoi-diagrams [44], and the construction of 2D and 3D medial axes. Graphics algorithms are readily available in common graphics APIs and many of the computations are performed by widespread graphics hardware.

D.3.5 Future work

- Extend these ideas to the case of curves.
- Study these ideas for constraining the epipolar geometry for less than 7 points in a RANSAC fashion.
- The ideas may have potential for computing other entities in multiple view geometry, such as trifocal tensor, auto-calibration equations, etc.
- Improve the performance for low levels of noise.
- Explicitly model point localization noise into the polygonal construction.

Bibliography

- [1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. In *ICCV '09: Proceedings of the Twelfth IEEE International Conference on Computer Vision*. IEEE Computer Society, 2009.
- [2] S. Alibhai and S. Zucker. Contour-based correspondence for stereo. In *Proc. Sixth European Conf. on Computer Vision*, Dublin, Ireland, jun 2000.
- [3] R.D. Arnold and T.O. Binford. Geometric constraints in stereo vision. In *Proc. SPIE*, volume 238–Image Processing for Missile Guidance, pages 281–292, San Diego, CA, 1980.
- [4] K. Astrom and A. Heyden. Multilinear constraints in the infinitesimal-time case. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 833–838, San Francisco, California, June 1996. IEEE Computer Society Press.
- [5] Kalle Astrom, Roberto Cipolla, and Peter Giblin. Generalised epipolar constraints. *Int. J. Comput. Vision*, 33(1):51–72, 1999.
- [6] Kalle Astrom and Fredrik Kahl. Motion estimation in image sequences using the deformation of apparent contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(2):114–127, 1999.
- [7] N. Ayache and B. Faverjon. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2):107–131, 1987.
- [8] N. Ayache and L. Lustman. Fast and reliable passive trinocular stereovision. In *1st International Conference on Computer Vision*, June 1987.
- [9] H. H. Baker and T. O. Binford. Depth from edge and intensity-based stereo. In *Proceedings of 7th International Joint Conferences on Artificial Intelligence*, pages 631–636, 1981.
- [10] Harry G. Barrow and Jay M. Tenenbaum. Interpreting linedrawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116, 1981.
- [11] B. Bascle and R. Deriche. Stereo matching, reconstruction and refinement of 3D curves using deformable contours. In *Proceedings of the Fourth International Conference on Computer Vision (Berlin, Germany, May 11–13, 1993)*, Washington, DC, 1993. IEEE Computer Society Press.

- [12] L. Baumela, L. Agapito, P. Bustos, and I. Reid. Motion estimation using the differential epipolar equation. *Proceedings of the 15th International Conference on Pattern Recognition*, 3:848–851, 2000.
- [13] Rikard Berthilsson, Kalle Åström, and Anders Heyden. Reconstruction of general curves, using factorization and bundle adjustment. *International Journal of Computer Vision*, 41(3):171–182, 2001.
- [14] Andrea Bottino and Aldo Laurentini. The visual hull of smooth curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1622–1632, 2004.
- [15] J. Y. Bouguet. Camera calibration toolbox for matlab, 2008.
- [16] Edmond Boyer and Marie-Odile Berger. 3D surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233, 1997.
- [17] A.T. Brint and M. Brady. Stereo matching of curves. *Image and Vision Computing*, 8(1):50–56, 1990.
- [18] T. Brodskỳ and C. Fermüller. Self-Calibration from Image Derivatives. *International Journal of Computer Vision*, 48(2):91–114, 2002.
- [19] Tomáš Brodskỳ, Cornelia Fermüller, and Yiannis Aloimonos. Structure from motion: Beyond the epipolar constraint. *Int. J. Comput. Vision*, 37(3):231–258, 2000.
- [20] Martin Bujnak, Zuzana Kukelova, and Tomáš Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *CVPR'08*. IEEE Computer Society, 2008.
- [21] R.L. Carceroni. *Recovering Non-Rigid 3D Motion, Shape and Reflectance from Multi-View Image Sequences: A Differential-Geometric Approach*. PhD thesis, University of Rochester, 2001.
- [22] R.L. Carceroni and K.N. Kutulakos. Toward recovering shape and motion of 3D curves from multi-view image sequences. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 23–25, Fort Collins, Colorado, USA, June 23-25 1999. IEEE Computer Society Press.
- [23] R.L. Carceroni and K.N. Kutulakos. Multi-view scene capture by Surfel sampling: From video streams to nonrigid 3d motion, shape and reflectance. *International Journal of Computer Vision*, 49(2):175–214, 2002.
- [24] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, November 1992.
- [25] Roberto Cipolla. *Active Visual Inference of Surface Shape*. PhD thesis, University of Oxford, 1991.

- [26] Roberto Cipolla, Kalle Åström, and Peter J. Giblin. Motion from the frontier of curved surfaces. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 269–275, Boston, Massachusetts, June 1995. IEEE Computer Society Press.
- [27] Roberto Cipolla and Peter Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, 1999.
- [28] Roberto Cipolla and Andrew Zisserman. Qualitative surface shape from deformation of image curves. *International Journal of Computer Vision*, 8(1):53–69, 1992.
- [29] Forrester Cole, Kevin Sanik, Doug DeCarlo, Adam Finkelstein, Thomas Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape? In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, August 2009.
- [30] D.B. Cooper, Y.P. Hung, and G. Taubin. A New Model-Based Stereo Approach For 3D Surface Reconstruction Using Contours On The Surface Pattern. In *Second International Conference on Computer Vision (Tampa,, FL, December 5–8, 1988)*, pages 906–911, Washington, D.C., 1988. IEEE Computer Society Press.
- [31] D.A. Cox, J.B. Little, and D. O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 1997.
- [32] D. Crispell, D. Lanman, P.G. Sibley, Y. Zhao, and G. Taubin. Shape from depth discontinuities. *LECTURE NOTES IN COMPUTER SCIENCE*, 5416, 2009.
- [33] Song De Ma. Conics-based stereo, motion estimation, and pose determination. *Int. J. Comput. Vision*, 10(1):7–25, 1993.
- [34] M. Ding, Y. Xiao, J. Peng, D. Schomburg, and B. Krebs. 3D reconstruction of free-formed line-like objects using NURBS representation. *Pattern Recognition*, 36(6):1255–1268, 2003.
- [35] Manfredo Perdigao do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, New Jersey, 1976.
- [36] F. Dornaika and A.D. Sappa. 3D Motion from Image Derivatives Using the Least Trimmed Square Regression? *LECTURE NOTES IN COMPUTER SCIENCE*, 4153:76, 2006.
- [37] Hossein Ebrahimnezhad and Hassan Ghassemian. Robust motion from space curves and 3D reconstruction from multiviews using perpendicular double stereo rigs. *Image and Vision Computing (In Press)*, 2008.
- [38] I. Eden and D. B. Cooper. Using 3D line segments for robust and efficient change detection from multiple noisy images. In *ECCV (2)*, volume 5303 of *Lecture Notes in Computer Science*. Springer, 2008.
- [39] J.H. Elder. Are edges incomplete? *International Journal of Computer Vision*, 34(2/3):1–23, August 1999.

- [40] C. Engels and D. Nistér. Global uncertainty in epipolar geometry via fully and partially data-driven sampling. In *Proceedings of ICCV ISPRS Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images, BenCOS*, 2005.
- [41] C. Engels, H. Stewénus, and D. Nistér. Bundle adjustment rules. In *Photogrammetric Computer Vision (PCV)*, SEP 2006.
- [42] W. Eric and L. Grimson. Computational experiments with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:17–33, 1985.
- [43] R. Fabbri, L. da F. Costa, J. C. Torelli, and O. M. Bruno. 2D Euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, 40(1):2:1–2:44, 2008.
- [44] R. Fabbri, L. da F. Costa, J. C. Torelli, and O. M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, 40(1):2:1–2:44, 2008.
- [45] Ricardo Fabbri, Peter J. Giblin, and Benjamin B. Kimia. Camera pose estimation using curve differential geometry. Technical report, LEMS – Brown University, 2010. to appear.
- [46] Ricardo Fabbri and Benjamin B. Kimia. High-order differential geometry of curves for multiview reconstruction and matching. In *Proceedings of the IEEE Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 645–660. Springer Verlag, November 2005.
- [47] Ricardo Fabbri and Benjamin B. Kimia. 3D curve sketch: Flexible curve-based stereo reconstruction and calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, California, USA, 2010. IEEE Computer Society Press.
- [48] O. Faugeras and T. Papadopoulo. A theory of the motion fields of curves. *International Journal of Computer Vision*, 10(2):125–156, 1993.
- [49] O.D. Faugeras and L. Robert. What can two images tell us about a third one? *International Journal of Computer Vision*, 18(1):5–19, April 1996.
- [50] Olivier Faugeras. *Three-Dimensional Computer Vision — A Geometric Viewpoint*. Artificial Intelligence. MIT Press, Cambridge, MA, USA, 1993.
- [51] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge, MA, USA, 2001.
- [52] Olivier D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? *ECCV92*, pages 563–578, 1992.
- [53] S. Finsterwalder and W. Scheufele. Das ruckwartseinschneiden im raum. *Sebastian Finsterwalder zum 75*, pages 86–100, 1937.
- [54] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

- [55] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.
- [56] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. In *CVPR'07*. IEEE Computer Society, 2007.
- [57] Yasutaka Furukawa and Jean Ponce. Carved visual hulls for image-based modeling. *Int. J. Comput. Vision*, 81(1):53–67, 2009.
- [58] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1362–1376, 2010.
- [59] Yasutaka Furukawa, Amit Sethi, Jean Ponce, and David J. Kriegman. Robust structure and motion from outlines of smooth curved surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(2):302–315, 2006.
- [60] P. J. Giblin and R. Weiss. Reconstruction of surfaces from profiles. In *First International Conference on Computer Vision, (London, England, June 8–11, 1987)*, pages 136–144, Washington, D.C., 1987. IEEE Computer Society Press.
- [61] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S.M. Seitz. Multi-view stereo for community photo collections. In *ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, pages 1–8. IEEE Computer Society, 2007.
- [62] W. E. L. Grimson. A Computer Implementation of a Theory of Human Stereo Vision. *Royal Society of London Philosophical Transactions Series B*, 292:217–253, May 1981.
- [63] J. A. Grunert. Das pothenotische problem in erweiterter gestalt nebst Über seine anwendungen in der geodäsie. *Archiv der für Mathematik and Physik*, 1:238–248, 1841.
- [64] Martin Habbecke and Leif Kobbelt. A surface-growing approach to multi-view stereo reconstruction. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007.
- [65] Robert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. Comput. Vision*, 13(3):331–356, 1994.
- [66] M. Harker and P. OLeary. Direct estimation of homogeneous vectors: An ill-solved problem in computer vision. *LECTURE NOTES IN COMPUTER SCIENCE*, 4338:919, 2006.
- [67] Chris Harris and Mike Stephens. A combined edge and corner detector. In *Alvey Vision Conference*, pages 189–192, 1988.
- [68] R. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1041, 1994.
- [69] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

- [70] R. I. Hartley. A linear method for reconstruction from lines and points. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 882–887, Boston, Massachusetts, June 1995. IEEE Computer Society Press.
- [71] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.
- [72] David J. Heeger and Allan D. Jepson. Subspace methods for recovering rigid motion i: algorithm and implementation. *Int. J. Comput. Vision*, 7(2):95–117, 1992.
- [73] C. Hernandez, F. Schmitt, and R. Cipolla. Silhouette coherence for camera calibration under circular motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):343–349, February 2007.
- [74] C. Hernández Esteban and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, december 2004.
- [75] Otto Hesse. Die cubische gleichung, von welcher die lösung des problems der homographie von m. chasles abhängt. *Journal reine angew. Math*, 62:188–192, 1863.
- [76] A. Heyden. Differential-Algebraic Multiview Constraints. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01*, pages 159–162. IEEE Computer Society Washington, DC, USA, 2006.
- [77] Radu Horaud, Bernard Conio, Olivier Le Boulleux, and Bernard Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics, and Image Processing*, 47(1):33–44, 1989.
- [78] Z. Y. Hu and F. C. Wu. A note on the number of solutions of the noncoplanar p4p problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):550–555, 2002.
- [79] Vishal Jain, Benjamin B. Kimia, and Joseph L. Mundy. Segregation of moving objects using elastic matching. *Computer Vision and Image Understanding*, 108:230–242, 2007.
- [80] Tanuja Joshi, Narendra Ahuja, and Jean Ponce. Structure and motion estimation from dynamic silhouettes under perspective projection. *International Journal of Computer Vision*, 31(1):31–50, 1999.
- [81] Michael Kaess, Rafal Zboinski, and Frank Dellaert. MCMC-based multiview reconstruction of piecewise smooth subdivision curves with a variable number of control points. In *ECCV'04*, volume 3021 of *Lecture Notes in Computer Science*, pages 329–341. Springer, 2004.
- [82] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. *International Journal of Computer Vision*, 74(1):3–15, 2007.
- [83] F. Kahl and A. Heyden. Euclidean reconstruction and auto-calibration from continuous motion. In *Proceedings of the Eighth International Conference on Computer Vision*, volume 2, Vancouver, Canada, July 9-12 2001. IEEE Computer Society Press.

- [84] Fredrik Kahl and Jonas August. Multiview reconstruction of space curves. In *Proceedings of the IEEE International Conference on Computer Vision*, page 1017, Washington, DC, USA, 2003.
- [85] Fredrik Kahl and Anders Heyden. Using conic correspondence in two images to estimate the epipolar geometry. In *Proceedings of the Sixth International Conference on Computer Vision*, page 761, Bombay, India, 1998. IEEE Computer Society Press.
- [86] J. Y. Kaminski and Amnon Shashua. Multiple view geometry of general algebraic curves. *International Journal of Computer Vision*, 56(3):195–219, February 2004.
- [87] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [88] J. J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.
- [89] Jan J. Koenderink. *Solid Shape*. MIT Press, Cambridge, Massachusetts, 1990.
- [90] Jan J. Koenderink and Andrea J. Van Doorn. The generic bilinear calibration-estimation problem. *Int. J. Comput. Vision*, 23(3):217–234, 1997.
- [91] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.
- [92] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:150–162, 1994.
- [93] Chi-Yin Lee. *An Integrated Approach to Stereo Matching and 3D Structure Reconstruction Using Image Contours*. PhD thesis, Brown University, Division of Engineering, 1993.
- [94] Gang Li and Steven W. Zucker. A differential geometrical model for contour-based stereo correspondence. In *Proc. IEEE Workshop on Variational, Geometric, and Level Set Methods in Computer Vision*, Nice, France, October 2003.
- [95] Gang Li and Steven W. Zucker. Contextual inference in contour-based stereo correspondence. *International Journal of Computer Vision*, 69(1):59–75, 2006.
- [96] Gang Li and Steven W. Zucker. Differential geometric inference in surface stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:72–86, 2010.
- [97] Shubao Liu and David Cooper. Ray markov random fields for image-based 3d modeling: Model and efficient inference. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, California, USA, 2009. IEEE Computer Society Press.

- [98] Shubao Liu, Kongbin Kang, Jean-Philippe Tarel, and David B. Cooper. Free-form object reconstruction from silhouettes, occluding edges and texture edges: A unified and robust operator based on duality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, march 2007.
- [99] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.
- [100] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. R. Society London*, 208(1173):385–397, July 1980.
- [101] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.
- [102] Q.-T. Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systemes robotiques*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, 1992.
- [103] Q.-T. Luong and O. D. Faugeras. On the determination of epipoles using cross-ratios. *Computer Vision and Image Understanding*, 71(1):1–18, 1998.
- [104] Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An invitation to 3D vision*. Springer, 2004.
- [105] C.R. Maurer, R. Qi, and V. Raghavan. A linear time algorithm for computing the Euclidean distance transform in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, feb 2003.
- [106] Stephen Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1992.
- [107] Stephen J. Maybank and Olivier D. Faugeras. A theory of self-calibration of a moving camera. *Int. J. Comput. Vision*, 8(2):123–151, 1992.
- [108] G.G. Medioni and R. Nevatia. Segment-based stereo matching. *CVGIP*, 31(1):2–18, July 1985.
- [109] Paulo R. S. Mendonça, Kwan-Yee K. Wong, and Roberto Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):604–616, 2001.
- [110] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.
- [111] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.
- [112] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. of the 5th International Joint Conference on Artificial Intelligence*, page 584, 1977.

- [113] Pierre Moreels and Pietro Perona. Evaluation of features detectors and descriptors based on 3d objects. *Int. J. Comput. Vision*, 73(3):263–284, 2007.
- [114] J. Mundy and A. Zisserman. Geometric invariance in computer vision. *MIT Press, Cambridge, USA*, 1992.
- [115] N. M. Nasrabadi. A stereo vision technique using curve-segments and relaxation matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(5):566–572, 1992.
- [116] S.J. Maybank O.D. Faugeras, Q.-T. Luong. Camera self-calibration: Theory and experiments. *ECCV92*, pages 321–334, 1992.
- [117] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.
- [118] J. Oliensis. Critique of structure-from-motion algorithms. *Computer Vision and Image Understanding*, 80(2):172–214, 2000.
- [119] Clark F. Olson and Daniel P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1):103–113, 1997.
- [120] Théo Papadopoulo. Motion analysis of 3d rigid curves from monocular image sequences. Technical Report RR-2779 (PhD Thesis), INRIA, jan 1996.
- [121] Théo Papadopoulo and Olivier Faugeras. Motion field of curves: applications. Technical Report RR-2168, INRIA, jan 1994.
- [122] Théodore Papadopoulo and Olivier D. Faugeras. Computing structure and motion of general 3D curves from monocular sequences of perspective images. In *Proceedings of the 4th European Conference on Computer Vision*, pages 696–708, London, UK, 1996. Springer-Verlag.
- [123] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: a stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.
- [124] T. Pollard, I. Eden, J. L. Mundy, and D. B. Cooper. A volumetric approach to change detection in satellite images. *ASPRS Photogrammetric Engineering & Remote Sensing Journal*, 2010. to appear.
- [125] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.
- [126] John Porrill and Stephen Pollard. Curve matching and stereo calibration. *Image and Vision Computing*, 9(1):45–50, 1991.
- [127] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 2nd edition edition, 1993. <http://www.nr.com/>.

- [128] Long Quan. Conic reconstruction and correspondence from two views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):151–160, Feb 1996.
- [129] Long Quan, Bill Triggs, and Bernard Mourrain. Some results on minimal euclidean reconstruction from four points. *J. Math. Imaging Vis.*, 24(3):341–348, 2006.
- [130] L. Reyes and E. Bayro Corrochano. The projective reconstruction of points, lines, quadrics, plane conics and degenerate quadrics using uncalibrated cameras. *Image and Vision Computing*, 23(8):693–706, August 2005.
- [131] L. Robert and O. D. Faugeras. Curve-based stereo: figural continuity and curvature. In *Proceedings of Computer Vision and Pattern Recognition*, pages 57–62, June 1991.
- [132] Jun Sato and Roberto Cipolla. Quasi-invariant parameterisations and matching of curves in images. *International Journal of Computer Vision*, 28(2):117–136, 1998.
- [133] Cordelia Schmid and Andrew Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233, 2000.
- [134] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR'06*, pages 519–528. IEEE Computer Society, 2006.
- [135] Ying Shan and Zhengyou Zhang. New measurements and corner-guidance for curve matching with probabilistic relaxation. *International Journal of Computer Vision*, 46(2):157–171, 2002.
- [136] Amnon Shashua. Trilinearity in visual recognition by alignment. In *Proceedings of the third European conference on Computer vision*, pages 479–484, Secaucus, NJ, USA, 1994. Springer-Verlag.
- [137] Yaser Sheikh, Asaad Hakeem, and Mubarak Shah. On the direct estimation of the fundamental matrix. In *CVPR'07*. IEEE Computer Society, 2007.
- [138] D. Sherman and S. Peleg. Stereo by incremental matching of contours. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(11):1102–1106, 1990.
- [139] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1270–1281, 2008.
- [140] Sudipta N. Sinha, Marc Pollefeys, and Leonard McMillan. Camera network calibration from dynamic silhouettes. In *CVPR'04*, pages 195–202. IEEE Computer Society, 2004.
- [141] Minas Spetsakis and John Yiannis Aloimonos. A multi-frame approach to visual motion perception. *Int. J. Comput. Vision*, 6(3):245–255, 1991.
- [142] R. Sturm. Das problem der projektivität und seine anwendung auf die flächen zweiten grades. *Math. Ann.*, 1:533–574, 1869.

- [143] Richard Szeliski and Richard Weiss. Robust shape recovery from occluding contours using a linear smoother. *Int. J. Comput. Vision*, 28(1):27–44, 1998.
- [144] Amir Tamrakar and Benjamin B. Kimia. No grouping left behind: From edges to curve fragments. In *ICCV '07: Proceedings of the Eleventh IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007. IEEE Computer Society.
- [145] Phil H. S. Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *Workshop on Vision Algorithms: Theory and Practice*. Springer-Verlag, 1999.
- [146] P.H.S. Torr and A.W. Fitzgibbon. Invariant fitting of two view geometry or in defiance of the 8 point algorithm? Technical Report MSR-TR-2002-50, Microsoft Research, may 2002.
- [147] B. Triggs. Differential Matching Constraints. In *Proceedings of the Fifth International Conference on Computer Vision*, Kerkyra, Greece, September 20-25 1999. IEEE Computer Society Press.
- [148] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment—A modern synthesis. *Vision Algorithms: Theory and Practice*, 1883:298–372, 2000.
- [149] Nhon H. Trinh and Benjamin B. Kimia. Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *International Journal of Computer Vision (IJCV)*, 2009, Under Review.
- [150] R. Vaillant and Olivier D. Faugeras. Using extremal boundaries for 3-d object modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):157–173, 1992.
- [151] A. van den Hengel. *Robust estimation of structure from motion in the uncalibrated case*. PhD thesis, Adelaide University, 2000.
- [152] A. van den Hengel, W. Chojnacki, and M.J. Brooks. Determining the Translational Speed of a Camera from Time-Varying Optical Flow. *LECTURE NOTES IN COMPUTER SCIENCE*, 3417:190, 2007.
- [153] Alessandro Verri and Alan Yuille. Some perspective projection invariants. *J. Opt. Soc. Am. A*, 5(3):426–431, 1988.
- [154] T. Vieville and OD Faugeras. The First Order Expansion of Motion Equations in the Uncalibrated Case. *Computer Vision and Image Understanding*, 64(1):128–146, 1996.
- [155] Allen M. Waxman and Shimon Ullman. Surface structure and three-dimensional motion from image flow kinematics. *International Journal of Robotics Research*, 4(3):72–94, 1985.
- [156] C. Wheatstone. Contributions to the physiology of vision – part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 128:371–394, 1838.
- [157] K. Y. Wong and R. Cipolla. Reconstruction of sculpture from its profiles with unknown camera positions. *IEEE Transactions on Image Processing*, 13(3):381–9, 2004.

- [158] Kwan-Yee Kenneth Wong, Paulo R. S. Mendonça, and Roberto Cipolla. Head model acquisition from silhouettes. In Carlo Arcelli, Luigi Cordella, and Gabriella Sanniti di Baja, editors, *IWVF*, pages 787–796, Capri, Italy, May 2001. Springer.
- [159] H. Wu and Y. Yu. Photogrammetric reconstruction of free-form objects with curvilinear structures. *The Visual Computer*, 21(4):203–216, 2005.
- [160] Yi Jun Xiao and Y. F. Li. Optimized stereo reconstruction of free-form space curves based on a nonuniform rational B-spline model. *J. Opt. Soc. Am. A*, 22(9):1746–1762, 2005.
- [161] Shankar Sastry Yi Ma, Jana Koseck. Motion recovery from image sequences: Discrete viewpoint vs. differential viewpoint. In *ECCV'98*, volume 1407 of *Lecture Notes in Computer Science*, page 337. Springer, 1998.
- [162] H. Zhang, K.Y.K. Wong, and G.Q.A. Zhang. Camera calibration from images of spheres. *Pattern Anal. and Mach. Intell.*, 29(3):499–502, March 2007.
- [163] J. Y. Zheng. Acquiring 3-d models from sequences of contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(2):163–178, 1994.
- [164] Steven W. Zucker. *Mathematical Models of Computer Vision: The Handbook*, chapter 22, pages 361–376. Differential Geometry from the Frenet Point of View: Boundary Detection, Stereo, Texture and Color. Springer, 2005. To appear.