Camera Pose Estimation Using First-Order Curve Differential Geometry

Ricardo Fabbri, Peter Giblin and Benjamin Kimia

Abstract—This paper considers and solves the problem of estimating camera pose given a pair of point-tangent correspondences between a 3D scene and a projected image. The problem arises when considering curve geometry as the basis of forming correspondences, computation of structure and calibration, which in its simplest form is a point augmented with the curve tangent. We show that while the resectioning problem is solved with a minimum of three points given the intrinsic parameters, when points are augmented with tangent information only two points are required, leading to substantial robustness and computational savings, *e.g.*, as a minimal engine within RANSAC. In addition, algorithms are developed to find a practical solution shown to effectively recover camera pose using synthetic and real datasets. This technology is intended as a building block of curve-based structure from motion systems, allowing new views to be incrementally registered to a core set of views for which relative pose has been computed.

Index Terms—Pose Estimation, Camera Resectioning, Differential Geometry

1 Introduction

A KEY problem in the reconstruction of structure from multiple views is the estimation of relative pose and intrinsic parameters among several cameras. The classical approach relies on corresponding points to incrementally determine parameters of each camera relative to initial pairs of relative pose solutions [1]. The correspondences can be determined from isolated keypoints such as SIFT [2] 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 correspondences using RANSAC [3] verified by the number of consistent features allow bootstrapping the reconstruction. This class of point-based methods is currently popular through packages such as OpenMVG, Bundler, and Colmap, large-scale 3D reconstuction [4], [5], [6], [6], [7], and Augmented Reality. However, drawbacks limit the applicability of keypoints, also noted in mainstream systems [8].

The correlation of interest points works under *limited baseline* [9], Fig. 1(a), while image curve fragments as underlying sharp ridges and reflectance curves persist stably over a much larger range of views. Moreover, keypoint technology depends on an abundance of features that survive the variations between views. While this is true in many scenes, in a non-trivial number of cases this is not, such as *(i)* Homogeneous regions, *e.g.*, from man-made objects [10], corridors, and cars [11], Fig. 1(b); *(ii)* Multiple moving objects require their own set of features which may not be sufficiently abundant without enough texture, Fig. 1(c); *(iii)* Non-rigid objects require a rich set of features per patch, Fig. 1(d). In all these cases there is often sufficient **image curve structure**, motivating a parallel technology employing image curves.

The use of image curves for auto-calibration (in the broad sense) can complement keypoint systems, but faces challenges that motivate the use of *small curve fragments*. First, edge linking does not generally produce results which persist stably across images, so that a curve in one view may become broken or grouped with other curve fragments, and while the underlying curve geometry correlates well across views, the individual curve fragments do not, Fig. 2(a-h). Second, even when curve fragments correspond, there is an intra-curve correspondence ambiguity, Fig. 2(i).

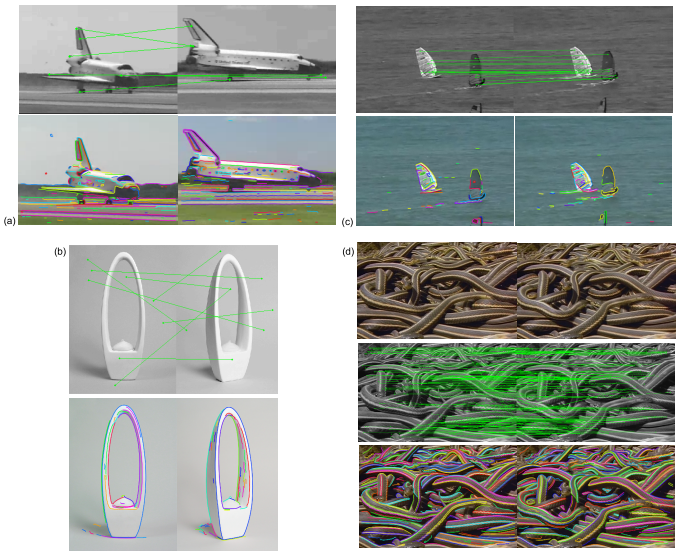

Fig. 1. (a) Widely separated views may not have enough keypoints in common, but share curve structure. (b) There may not be sufficient keypoints matching views of homogeneous objects such as a sculpture, but there is sufficient curve structure. (c) Each moving object requires its own set of features, demanding rich texture. (d) Non-rigid scenes face the same issue.

The paradigm explored in this paper is that small curve fragments, or points augmented with differential-geometric attributes, can be used as the basic image structure to correlate across views and perform structure from motion (SfM) [12], [13], [14], [15],

- *R. Fabbri is with the Department of Computational Modeling, Polytechnic Institute, State University of Rio de Janeiro. E-mail: see http://rfabbri.github.io*
- *P. Giblin is with the University of Liverpool. E-mail: pjgiblin@liverpool.ac.uk*
- *B. Kimia is with the School of Engineering, Brown University. E-mail: kimia@lems.brown.edu*

Manuscript received Month Day, Year; revised Month Day, Year.

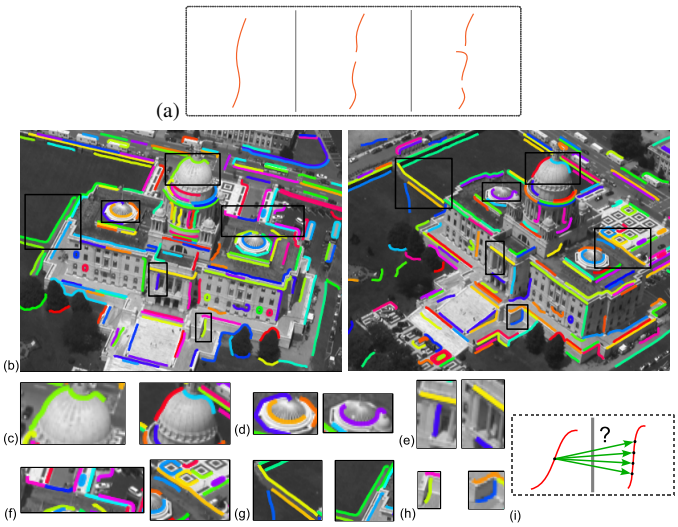

Fig. 2. Challenges in using curve fragments in multiview geometry: (a) instability with changes in viewpoint (b), zoom (c-h): a curve in one view broken in another, or linked to background, or becomes absent; a fragmentation at junctions gets linked in another view; different parts occluded in different views, and undergoing deformation from one view to the other. (i) Correspondence ambiguity along the curve.

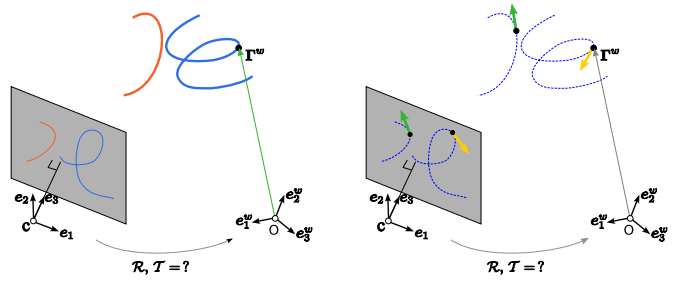

Fig. 3. Determining camera pose $\mathcal{R}$, $\mathcal{T}$ given space curves in a world coordinate system and their projections in an image coordinate system (left); and an approach consisting of determining pose given 3D point-tangents (local curve models) their projections (right).

[16], [17], [18]. Recent work [19], [20], [21] has shown that tangent, curvature, and the sign of curvature derivative can be reliably estimated. The value of this geometry is in correlating structure across three frames or more since the correspondence in two views is unconstrained. The differential geometry at two corresponding points in two views reconstruct the differential geometry of the space curve they arise from (possibly up to unknown parameters) and this constrains the differential geometry of corresponding curves in a third view [22], [23].

This paper explores the use of first-order differential geometry, namely points with tangent attributes, for determining the pose of a single camera. It proposes and solves the following:

Problem (P2PT) *For a camera with known intrinsic parameters, how many corresponding pairs of point-tangents in 3D space specified in world coordinates, and point-tangents in 2D specified in image coordinates, are required to establish the pose of the camera with respect to the world coordinates, Fig. 3?*

In principle, each 3D-2D correspondence yields three constraints in that projected coordinates x, y and direction must match the image measurements. Therefore, two correspondences should be necessary, and – generally – are sufficient for determining the 6 degrees of freedom of the pose. In this paper, we show that this is indeed true and provide a reference solver. Since the classical problem of determining camera pose from n 2D-3D correspondences is called Perspective-n-Point problem (PnP) – most notably P3P – we henceforth refer to the aforementioned problem as the P2PT problem, *i.e.*, Perspective-two-point-tangents.

The solution to P2PT is useful under several scenarios. In SfM, many views of the scene are to be reconstructed starting from an initial curve and/or oriented keypoint reconstruction available from two or three initial views [12], [13]. A curve can be obtained by edge detection and represented as a vector field of subpixel edgels, and keypoints can be corners, junctions, or SIFT points with the attributed dominant orientation, which have confirmed to be reliable for pose estimation since the suggestion from our initial work [24], [25]. A pair of point-tangents in the reconstruction can be matched with RANSAC to a pair of point-tangents in each new new image to be integrated to the initial reconstruction.

The advantage compared to using three correspondences from unorganized point reconstruction and P3P is that: *(i)* there are fewer edges than surface points (for a fixed discretization); *(ii)* the use of two rather than three points in RANSAC, requiring less than half the typical number of runs for the same level of robustness, *e.g.*, 32 vs. 70 runs to achieve 99.99% chance of hitting inlier correspondences in at least one run, assuming 50% outliers – though production systems often require as many runs as possible to maximize robustness. For the same number of runs N, P3P will fail orders of magnitude more often than P2PT, more so as N grows up to 100-1000, not uncommon in end-user systems requiring extreme robustness. This relative gap in robustness may be expressed as P3P having $(1-0.5^3)^N/(1-0.5^2)^N \approx 1.67^N$ times more failures per RANSAC solve than P2PT, where the numerator and denominator are the failure rates by hitting outliers for P3P and P2PT, resp [1]. Many more runs than the theoretical minimum are used in real systems, compounded for many views and datasets; and in fact the rate of outliers may be higher than expected, so this advantage accentuates.

Other useful scenarios for P2PT are: When a 3D model of an object is available from CAD or other sources, *e.g.*, vehicles, so a strategy similar to the first scenario to register new views can be used; In stereo video from calibrated binocular cameras, the reconstruction from one frame can be used to determine the pose in subsequent frames; When the extrinsic calibration of camera systems in indoor man-made scenes lacks reliable well-distributed points for P3P and bundle adjustment, *e.g.*, for precise metrology applications and in Augmented Reality where robustness is prime. The few available points are often at corners where long reliable lines meet; using tangents makes the most of scant data.

2 Related Work

Leveraging image curves for camera pose has generally relied on matching *epipolar tangencies* on *closed curves*. Corresponding points $\boldsymbol{\gamma}^1$ in image 1 and $\boldsymbol{\gamma}^2$ in image 2 are related by $\boldsymbol{\gamma}^{2\top} E \boldsymbol{\gamma}^1 = 0$, where E is the Essential matrix [26]; this can be extended to the differential geometry of two parametrized curves $\boldsymbol{\gamma}^1(s)$ and $\boldsymbol{\gamma}^2(s)$,

$$\boldsymbol{\gamma}^{1\top}(s) E \boldsymbol{\gamma}^2(s) = 0, \tag{2.1}$$

with unit tangents $\boldsymbol{t}^1(s)$ and $\boldsymbol{t}^2(s)$ related by differentiation

$$g^1(s) \boldsymbol{t}^{1\top}(s) E \boldsymbol{\gamma}^2(s) + \boldsymbol{\gamma}^{1\top}(s) E g^2(s) \boldsymbol{t}^2(s) = 0, \tag{2.2}$$

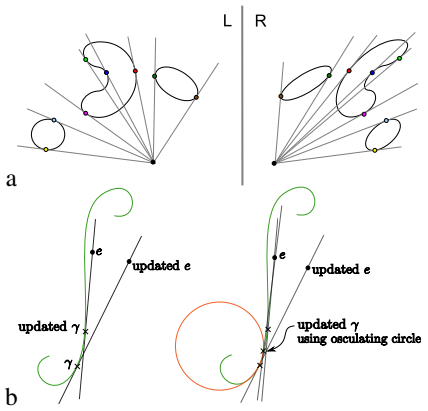

Fig. 4. Epipolar tangencies in curve-based relative pose estimation. **a** An epipolar line on the left must correspond to the epipolar line on the right having tangency on the corresponding curve. This works for both static curves and occluding contours. **b** Epipolar tangencies updated differentially through curvature.

where $g^1(s)$ and $g^2(s)$ are the respective speeds of parametrization of the curves. When one of the tangents $\boldsymbol{t}^1(s)$ is along the epipolar plane, $\boldsymbol{t}^{1\top}(s)E\boldsymbol{\gamma}^2(s) = 0$, then $\boldsymbol{\gamma}^{1\top}(s)E\boldsymbol{t}^2(s) = 0$. Thus, epipolar tangency in image 1 implies tangency in image 2, Fig. 4.

When the epipolar tangency constraint was first proposed [27], linked edges were employed with a coarse initial estimate of E to find a sparse set of epipolar tangencies, including at corners, in each view. They are manually matched between the views and used refine the estimate E, Fig. 4b, by minimizing $\boldsymbol{\gamma}^{1\top}(s)E\boldsymbol{\gamma}^2(s)$ over all matches in an iterative two-step scheme: the corresponding points are kept fixed and E is optimized in the first step 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 by the osculating circle. Closed curves were required.

The case of four corresponding conics in two views with unknown intrinsic parameters has been considered [28]. Each conic correspondence provides a pair of tangencies and two constraints, so four pairs of conics are needed. If intrinsic parameters are available, then the absolute conic gives two constraints on the epipolar geometry, so that only 3 conic correspondences are required. This, however, was deemed sensitive. Kaminski and Shashua [29] extended this to general algebraic curves in uncalibrated views, by extending Kruppa's equations to describe the epipolar constraint of two projections of a general algebraic curve.

Sinha *et. al.* [30] considered multiple static cameras viewing a moving object. Since the epipolar geometry between any pair of cameras is fixed, each hypothesized pair of epipoles representing a point in 4D is probed for a pair of epipolar tangencies across the video. 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, then optimized, providing intrinsic parameters and relative pose. Well-segmentable silhouettes were required.

There have also been recent developments on trifocal relative pose estimation to help bootstrap curve-based SfM systems [24]. This provides constraints from tangents unavailable in two views [22], but is relatively unexplored since it leads to high degree problems. The aim of the present work is to advance so-called "absolute" pose from curves, which despite the name is typically used to find the pose of new views *relative* to the frame of initial views that have been reconstructed by some other method, such as a trifocal or two-view solver. **Three** 2D-3D point correspondences are required to determine absolute pose [3], [31], [32], known as P3P or the *triangle pose problem*. This is an instance of the perspective n-point problem (PnP) [3], *i.e.*, the recovery of camera pose from n corresponding 3D-2D point pairs [33], or alternatively of depths [34], [35]. This can be seen as an intrinsically-calibrated case of the broader procedure of *camera resectioning* [1], [35], which in its general form assumes a camera matrix to be estimated (so that the intrinsic parameters can vary from solution to solution), and also related to *camera calibration* with the purpose of obtaining the intrinsic parameter matrix $\mathcal{K}_{im}$. In this paper, we deem all these variants *camera resectioning*.

Camera resectioning for unknown intrinsic parameters can be solved using **six** 3D–2D point correspondences when the intrinsic parameters are unknown, or **four** point correspondences when only the focal length is unknown, but all the other intrinsics are known [36], [37], Table 1. We show that when intrinsic parameters are known, only a pair of point-tangent correspondences are required to estimate pose. Work citing the conference version of this manuscript has since shown that 3 and 4 points, resp., are required for the other two cases [37], Table 1. This represents significant savings for a RANSAC computation. Notably, Kuang and Åström have done work on a similar relaxed problem [37]. In their context, point-tangents can be framed as "quivers" (*i.e.*, 1-quivers), or features with attributed directions (such as SIFT), but de-emphasizing the connection to tangents to general curves. We note that point-tangent fields can also be seen as vector fields, so that the results in this paper could potentially apply to surface-induced correspondence fields, or other vector fields.

The direct solution to P3P, given in 1841 [31], equates the sides of the triangle formed by the three points with those of the vectors in the image, which gives a system of three quadratics in three unknowns taken as the depths at each point. Following traditional methods going back to the German mathematician Grunert in 1841 [31] and later Finsterwalder in 1937 [32], by factoring out one depth, this can be reduced to a system of two quadratics in two unknowns – depth ratios of the second and third points relative to the first. Grunert further reduced this to a single quartic equation and Finsterwalder proposed an analytic solution. The state of the art P3P solver has been proposed very recently [38], reportedly achieving orders of magnitude advances in stability and speed. In the next section, **we show that only two points are necessary and sufficient to recover camera pose**, when tangent measurements or lines passing through the points are available. Similar to these seminal developments, we also reduce the equations to depth unknowns.

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

Notation: Our notation follows our paper [22], which contains a more practical and expressive version of the notation in the classic book by Cipolla and Giblin [16]. Compared to the standard Hartley and Zisserman book [1], the present notation has explicit scalings such as depths, and allows a clear expression of a differential theory (where limits of ratios are routinely considered), which is useful for relating the results to general curves, surfaces and singularity theory in the future. It fits within a larger curve-based SfM framework described in [14]. We now review the basics of this notation as to be self-contained. Consider a sequence of n

Case	Unknowns	Min. # of Point Corresp.	Min. # of Pt-Tgt Corresp.
Calibrated ($\mathcal{K}_{im}$ known)	Camera pose $\mathcal{R}$, $\mathcal{T}$	3	2 (this paper)
Focal length unknown	Pose $\mathcal{R}$, $\mathcal{T}$ and f	4	3
Uncalibrated ($\mathcal{K}_{im}$ unknown)	Camera model $\mathcal{K}_{im}$, $\mathcal{R}$, $\mathcal{T}$	6	4

TABLE 1
The number of 3D–2D correspondences needed to solve for camera pose and intrinsic parameters.

3D points $(\boldsymbol{\Gamma}_1^w, \boldsymbol{\Gamma}_2^w, \ldots, \boldsymbol{\Gamma}_n^w)$, described in the world coordinate system (hence the w superscript) and their corresponding projected image points $(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_n)$ described as points in the 3D camera coordinate system as 3-vectors with third coordinate 1. We may drop the subscript when an expression holds for any index. Let the rotation $\mathcal{R}$ and translation $\mathcal{T}$ relate the camera and world coordinate systems through

$$\boldsymbol{\Gamma} = \mathcal{R}\boldsymbol{\Gamma}^w + \mathcal{T}, \tag{3.1}$$

where $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}^w$ are the coordinates of a point in the camera and world coordinate systems, respectively. Let $\rho_1, \rho_2, \ldots, \rho_n$ be depths defined by the projection equation

$$\boldsymbol{\Gamma} = \rho\boldsymbol{\gamma}. \tag{3.2}$$

In general we assume that each point $\boldsymbol{\gamma}_i$ is a sample from an image curve $\boldsymbol{\gamma}_i(s_i)$ which is the projection of a space curve $\boldsymbol{\Gamma}_i(S_i)$ represented in the camera coordinate system, where s_i and S_i are arclengths along the image and space curves, resp. In terms of the usual notation [1] based on a 3×4 camera matrix, we have

$$\boldsymbol{\gamma}_{im} \;\propto\; \mathcal{K}_{im}\boldsymbol{\gamma} \;\propto\; \mathcal{K}_{im}\boldsymbol{\Gamma} \;\propto\; \mathcal{K}_{im} \cdot [\mathcal{R} \,|\, \mathcal{T}] \begin{bmatrix} \boldsymbol{\Gamma}^w \\ 1 \end{bmatrix}, \tag{3.3}$$

where $\boldsymbol{\gamma}_{im}$ is a point in pixel image coordinates, $\mathcal{K}_{im}$ is the 3×3 calibration matrix [1], and $\mathcal{K}_{im} \cdot [\mathcal{R} \,|\, \mathcal{T}]$ the usual camera matrix.

We consider tangent measurements $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n$ and their corresponding 3D tangents $\boldsymbol{T}_1^w, \boldsymbol{T}_2^w, \ldots \boldsymbol{T}_n^w$ are given, and will show that $n = 2$ is suffices. We take the 2D-3D point-tangents as samples along 2D-3D curve neighborhoods, respectively, where the speed of parametrization along the image curve neighborhoods are g_i and along the space curves G_i, $i = 1, 2, \ldots, n$. Each tangent can be taken as a unit vector obtained by differentiating some paramatrization of the underlying curve: $\boldsymbol{T}^w = \frac{\boldsymbol{\Gamma}^{w\prime}}{\|\boldsymbol{\Gamma}^{w\prime}\|}$, and $\boldsymbol{t} = \frac{\boldsymbol{\gamma}'}{\|\boldsymbol{\gamma}'\|}$, where prime "′" denotes differentiation with respect to an arbitrary parameter, for all i. Note that since $\boldsymbol{\gamma}$'s have third coordinate 1, the tangent vectors $\boldsymbol{t}$'s have third coordinate zero. These tangents can also be seen as infinite lines going through the points $\boldsymbol{\Gamma}_i^w$ and $\boldsymbol{\gamma}_i$, although we also exploit orientation constraints.

For completeness, we recall certain multiview relations regarding tangent vectors [22]. The tangent vector in camera coordinates is $\boldsymbol{T} = \mathcal{R}\boldsymbol{T}^w$, a consequence of differentiating Eq. 3.1 with respect to an arbitrary curve parameter and the fact that $\mathcal{R}$ is length-preserving. Corresponding 3D-2D tangents $\boldsymbol{t}$ and $\boldsymbol{T}^w$ are related by also differentiating the projection Equation 3.2 with respect to a common (synchronized) curve parameter

$$G\boldsymbol{T} = G\mathcal{R}\boldsymbol{T}^w = \rho'\boldsymbol{\gamma} + \rho g\boldsymbol{t}, \tag{3.4}$$

where ρ' is the depth derivative with respect to such parameter. In particular, this common parameter can often be taken as the arc length of the space curve $G = 1$ or of the image curve $g = 1$. This equation states that $\boldsymbol{T}$, $\boldsymbol{t}$, and $\boldsymbol{\gamma}$ are coplanar and is central to the present paper. From this, 3D tangents can be projected $\boldsymbol{T} \longrightarrow \boldsymbol{t}$,

$$\boldsymbol{t} = \frac{\boldsymbol{T} - T_z\boldsymbol{\gamma}}{\|\boldsymbol{T} - T_z\boldsymbol{\gamma}\|}, \tag{3.5}$$

and reconstructed from 2D tangents $\boldsymbol{t}$ observed in two views [22, equation 4.11].

Theorem 3.1. *Given a pair of 3D point-tangents* $\{(\boldsymbol{\Gamma}_1^w, \boldsymbol{T}_1^w), (\boldsymbol{\Gamma}_2^w, \boldsymbol{T}_2^w)\}$ *described in a world coordinate system and their corresponding perspective projections, the 2D point-tangents* $(\boldsymbol{\gamma}_1, \boldsymbol{t}_1)$, $(\boldsymbol{\gamma}_2, \boldsymbol{t}_2)$, *the pose of the camera* $\mathcal{R}$, $\mathcal{T}$ *relative to the world coordinate system defined by* $\boldsymbol{\Gamma} = \mathcal{R}\boldsymbol{\Gamma}^w + \mathcal{T}$ *can be solved up to a finite number of solutions (assuming that the intrinsic parameters* $\mathcal{K}_{im}$ *are known), by solving the system*

$$\begin{cases} \boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1\,\rho_1^2 - 2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2\,\rho_1\rho_2 + \boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2\,\rho_2^2 = \|\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w\|^2, \\ Q(\rho_1, \rho_2) = 0, \end{cases} \tag{3.6}$$

where $\mathcal{R}\boldsymbol{\Gamma}_1^w + \mathcal{T} = \boldsymbol{\Gamma}_1 = \rho_1\boldsymbol{\gamma}_1$ *and* $\mathcal{R}\boldsymbol{\Gamma}_2^w + \mathcal{T} = \boldsymbol{\Gamma}_2 = \rho_2\boldsymbol{\gamma}_2$, *and* $Q(\rho_1, \rho_2)$ *is an eight degree polynomial. This then solves for* $\mathcal{R}$ *and* $\mathcal{T}$ *as*

$$\begin{cases} \mathcal{R} = \begin{bmatrix} (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w) & \boldsymbol{T}_1^w & \boldsymbol{T}_2^w \end{bmatrix}^{-1} \cdot \\ \qquad \begin{bmatrix} \rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2 & \rho_1\frac{g_1}{G_1}\boldsymbol{t}_1 + \frac{\rho_1'}{G_1}\boldsymbol{\gamma}_1 & \rho_2\frac{g_2}{G_2}\boldsymbol{t}_2 + \frac{\rho_2'}{G_2}\boldsymbol{\gamma}_2 \end{bmatrix} \\ \mathcal{T} = \rho_1\boldsymbol{\gamma}_1 - \mathcal{R}\boldsymbol{\Gamma}_1^w, \end{cases}$$

where expressions for four auxiliary variables $\frac{g_1}{G_1}$ *and* $\frac{g_2}{G_2}$, *the ratio of speeds in the image and along the tangents, and* ρ_1 *and* ρ_2 *are available.*

Proof. The proof proceeds by *(i)* writing the projection equations for each point and its 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. This gives six equations in six unknowns: $(\rho_1, \rho_2, \rho_1\frac{g_1}{G_1}, \rho_2\frac{g_2}{G_2}, \frac{\rho_1'}{G_1}, \frac{\rho_2'}{G_2})$; *(iv)* eliminating the unknowns ρ_1' and ρ_2' gives four quadratic equations in four unknowns: $(\rho_1, \rho_2, \rho_1\frac{g_1}{G_1}, \rho_2\frac{g_2}{G_2})$. Three of these quadratics can be written in the form:

$$\begin{cases} Ax_1^2 + Bx_1 + C = 0 & (3.7) \\ Ex_2^2 + Fx_2 + G = 0 & (3.8) \\ H + Jx_1 + Kx_2 + Lx_1x_2 = 0, & (3.9) \end{cases}$$

where $x_1 = \rho_1\frac{g_1}{G_1}$ and $x_2 = \rho_2\frac{g_2}{G_2}$ and where A through L are only functions of the two unknowns ρ_1 and ρ_2. Now, Eq. 3.9 represents a rectangular hyperbola, Fig. 5, while Eqs. 3.7 and 3.8 vertical and horizontal lines in the (x_1, x_2) space. Fig. 5 illustrates that only one solution is possible which is then analytically written in terms of variables A–L (not shown here). This allows expressing $\rho_1\frac{g_1}{G_1}$ and $\rho_2\frac{g_2}{G_2}$ in terms of ρ_1 and ρ_2 – a degree 16 polynomial – but this is in fact divisible by $\rho_1^4\rho_2^4$, leaving a polynomial Q of degree

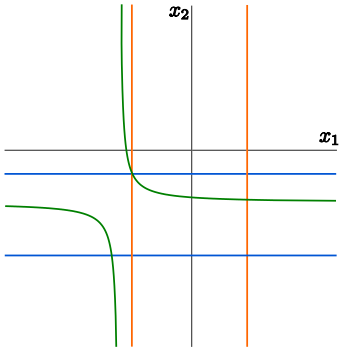

Fig. 5. Diagram of the mutual intersection of Eqs. 3.7–3.9 in the x_1–x_2 plane.

8. Furthermore, we find that $Q(-\rho_1, -\rho_2) = Q(\rho_1, \rho_2)$, using the symmetry of the original equations. This, together with the unused equation (the remaining one of four) gives the system 3.6. For clarity, the details of the proof are completed at the end of this section. □

Proposition 3.2. *The algebraic solutions to the system (3.6) of Theorem 3.1 are also required to satisfy the following inequalities arising from imaging and other requirements enforced by*

$$\rho_1 > 0, \;\; \rho_2 > 0 \tag{3.10}$$

$$\frac{g_1}{G_1} > 0, \;\; \frac{g_2}{G_2} > 0 \tag{3.11}$$

$$\frac{\det[\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2 \;\; \rho_1\frac{g_1}{G_1}\boldsymbol{t}_1 + \frac{\rho_1'}{G_1}\boldsymbol{\gamma}_1 \;\; \rho_2\frac{g_2}{G_2}\boldsymbol{t}_2 + \frac{\rho_2'}{G_2}\boldsymbol{\gamma}_2]}{\det\left[\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w \;\; \boldsymbol{T}_1^w \;\; \boldsymbol{T}_2^w\right]} > 0. \tag{3.12}$$

Proof. There are multiple solutions for ρ_1 and ρ_2 in Eq. 3.6. Observe that if ρ_1, ρ_2, $\mathcal{R}$, $\mathcal{T}$ are a solution, then so are $-\rho_1$, $-\rho_2$, $-\mathcal{R}$, and $-\mathcal{T}$. Only one of these two solutions are valid, as the camera geometry enforces positive depth, $\rho_1 > 0$ and $\rho_2 > 0$; 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.

Second, observe that the matrix $\mathcal{R}$ can only be a rotation matrix if it has determinant $+1$ and is a reflection if it has determinant -1. Using (3.7), $\det(\mathcal{R})$ can be written as

$$\det\mathcal{R} = \frac{\det\left[\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2 \;\; \rho_1\frac{g_1}{G_1}\boldsymbol{t}_1 + \frac{\rho_1'}{G_1}\boldsymbol{\gamma}_1 \;\; \rho_2\frac{g_2}{G_2}\boldsymbol{t}_2 + \frac{\rho_2'}{G_2}\boldsymbol{\gamma}_2\right]}{\det\left[\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w \;\; \boldsymbol{T}_1^w \;\; \boldsymbol{T}_2^w\right]}.$$

Finally, the space curve tangent $\boldsymbol{T}$ and the image curve tangent $\boldsymbol{t}$ must point in the same direction: $\boldsymbol{T}\cdot\boldsymbol{t} > 0$, or $\frac{g_1}{G_1} > 0$ and $\frac{g_2}{G_2} > 0$. This is because the parametrization we have assumed in the space curve projects $\boldsymbol{T}$ to the same half plane as $\boldsymbol{t}$ in each view so that $\boldsymbol{T}$ and $\boldsymbol{t}$ need to point in the same direction, *i.e.*, $\boldsymbol{T}\cdot\boldsymbol{t} > 0$, or from Eqs. 3.23 and 3.24, $\frac{g_1}{G_1} > 0$ and $\frac{g_2}{G_2} > 0$. □

3.1 Completing the proof of Theorem 3.1

In the course of proving Theorem 3.1 in further detail, we will *also* show that

Lemma 3.3. *The polynomial $Q(\rho_1, \rho_2)$ whose zero-locus describes the possible depths of given corresponding 3D point-tangents and 2D point-tangents from (3.6) is given by*

$$\begin{aligned} &Q(\rho_1, \rho_2) = \\ &A^3M^2 + AC^2N^2 - 2A^2CMN \\ &+ (BCN - ABM)(AP - BN) + C(AP - BN)^2 = 0 \end{aligned} \tag{3.13}$$

where the parameters A through P are defined as

$$\begin{cases} A = 1 - 2\boldsymbol{\gamma}_1^\top\boldsymbol{t}_1B_1 + \boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1B_1^2 \\ B = [2(\boldsymbol{\gamma}_1^\top\boldsymbol{t}_1) - 2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1B_1]A_1 \\ C = (\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1)A_1^2 - 1 \\ E = 1 - 2\boldsymbol{\gamma}_2^\top\boldsymbol{t}_2B_2 + \boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2B_2^2 \\ F = [2(\boldsymbol{\gamma}_2^\top\boldsymbol{t}_2) - 2\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2B_2]A_2 \\ G = (\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)A_2^2 - 1 \\ H = \boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2A_1A_2 - (\boldsymbol{T}_1^w)^\top\boldsymbol{T}_2^w \\ J = [\boldsymbol{\gamma}_2^\top\boldsymbol{t}_1 - \boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2B_1]A_2 \\ K = [\boldsymbol{\gamma}_1^\top\boldsymbol{t}_2 - \boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2B_2]A_1 \\ L = \boldsymbol{t}_1^\top\boldsymbol{t}_2 - \boldsymbol{\gamma}_2^\top\boldsymbol{t}_1B_2 - \boldsymbol{\gamma}_1^\top\boldsymbol{t}_2B_1 + \boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2B_1B_2, \\ M = EH^2 - FHK + GK^2 \\ N = EJ^2 - FJL + GL^2 \\ P = (2EHJ - FHL - FJK + 2GKL) \end{cases} \tag{3.14}$$

where

$$\begin{cases} A_1 = \dfrac{(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top\boldsymbol{T}_1^w}{(\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2)^\top\boldsymbol{\gamma}_1} \\ A_2 = \dfrac{(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top\boldsymbol{T}_2^w}{(\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2)^\top\boldsymbol{\gamma}_2} \\ B_1 = \dfrac{(\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2)^\top\boldsymbol{t}_1}{(\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2)^\top\boldsymbol{\gamma}_1} \\ B_2 = \dfrac{(\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2)^\top\boldsymbol{t}_2}{(\rho_1\boldsymbol{\gamma}_1 - \rho_2\boldsymbol{\gamma}_2)^\top\boldsymbol{\gamma}_2}. \end{cases} \tag{3.15}$$

Moreover,

$$\begin{cases} \rho_1\dfrac{g_1}{G_1} = -\dfrac{AM - CN}{AP - BN} \\ \rho_2\dfrac{g_2}{G_2} = -\dfrac{EM_A - GN_A}{EP_A - FN_A}, \end{cases} \tag{3.16}$$

where M_A, N_A, and P_A are defined as

$$\begin{cases} M_A = AH^2 - BHJ + CJ^2 \\ N_A = AK^2 - BKL + CL^2 \\ P_A = 2AHK - BHL - BKJ + 2CJL, \end{cases} \tag{3.17}$$

and

$$\begin{cases} \dfrac{\rho_1'}{G_1} = A_1 - B_1\rho_1\dfrac{g_1}{G_1} \\ \dfrac{\rho_2'}{G_2} = A_2 - B_2\rho_2\dfrac{g_2}{G_2}. \end{cases} \tag{3.18}$$

Proof. (*Of the Lemma and completing that of Theorem 3.1*) An image point $\boldsymbol{\gamma}$ is related to the underlying space point $\boldsymbol{\Gamma}$ through $\boldsymbol{\Gamma} = \rho\boldsymbol{\gamma}$, Eq. 3.2. A space point $\boldsymbol{\Gamma}$ in local coordinates is related to $\boldsymbol{\Gamma}^w$ in the world coordinates by a rotation matrix $\mathcal{R}$ and translation $\mathcal{T}$ through Eq. 3.1. Equating these at each of the two points gives

$$\begin{cases} \rho_1\boldsymbol{\gamma}_1 = \mathcal{R}\boldsymbol{\Gamma}_1^w + \mathcal{T} \\ \rho_2\boldsymbol{\gamma}_2 = \mathcal{R}\boldsymbol{\Gamma}_2^w + \mathcal{T}, \end{cases} \tag{3.19}$$

where ρ_1 and ρ_2 are the depth at image points $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$, respectively. By differentiating with respect to the parameters of $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$ we have:

$$\begin{cases} \rho_1 g_1 \boldsymbol{t}_1 + \rho_1' \boldsymbol{\gamma}_1 = \mathcal{R} G_1 \boldsymbol{T}_1^w \\ \rho_2 g_2 \boldsymbol{t}_2 + \rho_2' \boldsymbol{\gamma}_2 = \mathcal{R} G_2 \boldsymbol{T}_2^w, \end{cases} \tag{3.20}$$

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 $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$, respectively, and G_1 and G_2 are the speeds of parametrization of the space curves $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$, respectively. The vector Equations 3.19 and 3.20 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, but we keep them in general form. The given quantities are $\boldsymbol{\gamma}$, $\boldsymbol{t}$, and $\boldsymbol{\Gamma}^w$, $\boldsymbol{T}^w$ 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 Eqs. (3.19)

$$\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2 = \mathcal{R}(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w), \tag{3.21}$$

which together with Eq. 3.20 gives a system of equations

$$\begin{cases} \rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2 = \mathcal{R}(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w) & (3.22) \\ \rho_1 \dfrac{g_1}{G_1} \boldsymbol{t}_1 + \dfrac{\rho_1'}{G_1} \boldsymbol{\gamma}_1 = \mathcal{R} \boldsymbol{T}_1^w & (3.23) \\ \rho_2 \dfrac{g_2}{G_2} \boldsymbol{t}_2 + \dfrac{\rho_2'}{G_2} \boldsymbol{\gamma}_2 = \mathcal{R} \boldsymbol{T}_2^w. & (3.24) \end{cases}$$

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 (3.22)–(3.24). The number of unknowns can be reduced by eliminating $\mathcal{R}$ in a second step. The matrix $\mathcal{R}$ rotates three known vectors, $(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)$, $\boldsymbol{T}_1^w$, and $\boldsymbol{T}_2^w$ 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 (3.22)–(3.24), *i.e.*,

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

This is a system of six equations. Note that a clear geometric condition for the problem to have a solution is that the vectors $\{(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w), \boldsymbol{T}_1^w, \boldsymbol{T}_2^w\}$ be non-coplanar. Taking the product of the left hand matrix (also the right hand matrix) with its transpose, and using $\mathcal{R}^\top \mathcal{R} = I$, gives

$$\begin{cases} (\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top (\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2) = (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w) \\ (\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top (\rho_1 \frac{g_1}{G_1} \boldsymbol{t}_1 + \frac{\rho_1'}{G_1} \boldsymbol{\gamma}_1) = (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top \boldsymbol{T}_1^w \\ (\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top (\rho_2 \frac{g_2}{G_2} \boldsymbol{t}_2 + \frac{\rho_2'}{G_2} \boldsymbol{\gamma}_1) = (\boldsymbol{\Gamma}_2^w - \boldsymbol{\Gamma}_2^w)^\top \boldsymbol{T}_2^w \\ (\rho_1 \frac{g_1}{G_1} \boldsymbol{t}_1 + \frac{\rho_1'}{G_1} \boldsymbol{\gamma}_1)^\top (\rho_1 \frac{g_1}{G_1} \boldsymbol{t}_1 + \frac{\rho_1'}{G_1} \boldsymbol{\gamma}_1) = 1 \\ (\rho_2 \frac{g_2}{G_2} \boldsymbol{t}_2 + \frac{\rho_2'}{G_2} \boldsymbol{\gamma}_2)^\top (\rho_2 \frac{g_2}{G_2} \boldsymbol{t}_2 + \frac{\rho_2'}{G_2} \boldsymbol{\gamma}_2) = 1 \\ (\rho_1 \frac{g_1}{G_1} \boldsymbol{t}_1 + \frac{\rho_1'}{G_1} \boldsymbol{\gamma}_1)^\top (\rho_2 \frac{g_2}{G_2} \boldsymbol{t}_2 + \frac{\rho_2'}{G_2} \boldsymbol{\gamma}_2) = (\boldsymbol{T}_1^w)^\top \boldsymbol{T}_2^w. \end{cases} \tag{3.25}$$

The first equation is a quadratic in ρ_1 and ρ_2

$$\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 \, \rho_1^2 - 2 \boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_2 \, \rho_1 \rho_2 + \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2 \, \rho_2^2 = (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w), \tag{3.26}$$

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

$$(\boldsymbol{\gamma}_1 \cdot \boldsymbol{\gamma}_2)^2 - (\boldsymbol{\gamma}_1 \cdot \boldsymbol{\gamma}_1)(\boldsymbol{\gamma}_2 \cdot \boldsymbol{\gamma}_2) = -\|\boldsymbol{\gamma}_1 \times \boldsymbol{\gamma}_2\|^2 < 0 \tag{3.27}$$

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 $\rho_1^2 \|\boldsymbol{\gamma}_1\|^2 = \|\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w\|^2$, or real roots $\rho_1 = \pm \frac{\|\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w\|}{\|\boldsymbol{\gamma}_1\|}$.

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

$$\begin{cases} [(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{\gamma}_1] \dfrac{\rho_1'}{G_1} \\ \qquad = (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top \boldsymbol{T}_1^w - [(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{t}_1] \rho_1 \dfrac{g_1}{G_1} \\ [(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{\gamma}_2] \dfrac{\rho_2'}{G_2} \\ \qquad = (\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top \boldsymbol{T}_2^w - [(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{t}_2] \rho_2 \dfrac{g_2}{G_2}, \end{cases} \tag{3.28}$$

or

$$\begin{cases} \dfrac{\rho_1'}{G_1} = \dfrac{(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top \boldsymbol{T}_1^w}{(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{\gamma}_1} - \dfrac{(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{t}_1}{(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{\gamma}_1} \cdot \rho_1 \dfrac{g_1}{G_1} \\ \qquad = A_1 - B_1 \rho_1 \dfrac{g_1}{G_1} \\ \dfrac{\rho_2'}{G_2} = \dfrac{(\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w)^\top \boldsymbol{T}_2^w}{(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{\gamma}_2} - \dfrac{(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{t}_2}{(\rho_1 \boldsymbol{\gamma}_1 - \rho_2 \boldsymbol{\gamma}_2)^\top \boldsymbol{\gamma}_2} \cdot \rho_2 \dfrac{g_2}{G_2} \\ \qquad = A_2 - B_2 \rho_2 \dfrac{g_2}{G_2}, \end{cases} \tag{3.29}$$

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 (3.25) can be expanded as

$$\begin{cases} \left(\rho_1 \dfrac{g_1}{G_1}\right)^2 + 2 \boldsymbol{\gamma}_1^\top \boldsymbol{t}_1 \left(\rho_1 \dfrac{g_1}{G_1}\right) \left(\dfrac{\rho_1'}{G_1}\right) + \boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 \left(\dfrac{\rho_1'}{G_1}\right)^2 = 1 \\ \left(\rho_2 \dfrac{g_2}{G_2}\right)^2 + 2 \boldsymbol{\gamma}_2^\top \boldsymbol{t}_2 \left(\rho_2 \dfrac{g_2}{G_2}\right) \left(\dfrac{\rho_2'}{G_2}\right) + \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2 \left(\dfrac{\rho_2'}{G_2}\right)^2 = 1 \\ \boldsymbol{t}_1^\top \boldsymbol{t}_2 \, \rho_1 \dfrac{g_1}{G_1} \cdot \rho_2 \dfrac{g_2}{G_2} + \boldsymbol{\gamma}_2^\top \boldsymbol{t}_1 \, \rho_1 \dfrac{g_1}{G_1} \cdot \dfrac{\rho_2'}{G_2} + \boldsymbol{\gamma}_1^\top \boldsymbol{t}_2 \rho_2 \dfrac{g_2}{G_2} \cdot \dfrac{\rho_1'}{G_1} + \\ \qquad \boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_2 \dfrac{\rho_1'}{G_1} \cdot \dfrac{\rho_2'}{G_2} = (\boldsymbol{T}_1^w)^\top \boldsymbol{T}_2^w. \end{cases}$$

Substituting $\frac{\rho_1'}{G_1}$ and $\frac{\rho_2'}{G_2}$ from Eqs. 3.29 gives

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

These three equations can be written in summary form using $x_1 = \rho_1 \frac{g_1}{G_1}$ and $x_2 = \rho_2 \frac{g_2}{G_2}$, Equations 3.7–3.9, with A through L functions only of the two unknowns ρ_1 and ρ_2. Thus, these three equations after solving for x_1 and x_2 express a relationship between ρ_1 and ρ_2, which together with Eq. 3.26 can lead to a solution for ρ_1 and ρ_2.

Equation 3.9, with given values for ρ_1 and ρ_2, represents a rectangular hyperbola in the x_1–x_2 plane, as illustrated in Fig. 5, and each of the Eqs. 3.7 and 3.8 represents a pair of (real) lines in the same plane, parallel respectively to the x_2 and x_1 axes. In general there will not be more than one intersection between the aforementioned curves.

Specifically, the variables x_1 and x_2 can be solved by rewriting Eq. 3.9 as

$$(H + Jx_1) + (K + Lx_1)x_2 = 0, \tag{3.30}$$

giving

$$x_2 = -\frac{H + Jx_1}{K + Lx_1}. \tag{3.31}$$

Using this expression in Eq. 3.8 gives

$$E\frac{(H + Jx_1)^2}{(K + Lx_1)^2} - F\frac{H + Jx_1}{K + Lx_1} + G = 0, \tag{3.32}$$

or

$$E(H + Jx_1)^2 - F(H + Jx_1)(K + Lx_1) + G(K + Lx_1)^2 = 0. \tag{3.33}$$

Reorganizing as a quadratic in x_1, this solves for x_1 which together with Eq. 3.7 gives a constraint on the parameters depending on ρ_1 and ρ_2,

$$\begin{cases} (EJ^2 - FJL + GL^2)x_1^2 & (3.34) \\ \quad + (2EHJ - FHL - FJK + 2GKL)x_1 & (3.35) \\ \quad + (EH^2 - FHK + GK^2) = 0 \\ Ax_1^2 + Bx_1 + C = 0. & (3.36) \end{cases}$$

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

$$(AP - BN)x_1 + AM - CN = 0, \tag{3.37}$$

so that

$$x_1 = -\frac{AM - CN}{AP - BN}. \tag{3.38}$$

Substituting back into Eq. 3.36 gives

$$A\left[\frac{AM - CN}{AP - BN}\right]^2 + -B\frac{AM - CN}{AP - BN} + C = 0, \tag{3.39}$$

or

$$\begin{aligned} &A^3M^2 + AC^2N^2 - 2A^2CMN \\ &+ (BCN - ABM)(AP - BN) + C(AP - BN)^2 = 0 \end{aligned} \tag{3.40}$$

The equation, after expressions for $A, B, \ldots, P$ are substituted in, can be divided by $\rho_1^4\rho_2^4$, giving an 8^{th} order polynomial equation in ρ_1 and ρ_2, *i.e.*, $Q(\rho_1, \rho_2) = 0$. This equation together with Eq. 3.26 represents a system of two equations in two unknowns

$$\begin{cases} \boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 \rho_1^2 - 2\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_2 \rho_1\rho_2 + \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2 \rho_2^2 = \|\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w\|^2, \\ Q(\rho_1, \rho_2) = 0, \end{cases} \tag{3.41}$$

and gives a number of solutions for ρ_1, and ρ_2 which in turn 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 (3.1). The translation $\mathcal{T}$ is then solved from Eqs. 3.19 as

$$\mathcal{T} = \rho_1 \boldsymbol{\gamma}_1 - \mathcal{R}\boldsymbol{\Gamma}_1^w. \tag{3.42}$$

□

4 A Practical Approach Towards a Solver

Equations 3.6 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 shown to be an ellipse, it is possible to parametrize it by a bracketed parameter and then look for intersections with the second curve which is of degree 8. This gives a higher-order polynomial in a *single* unknown which can be solved more readily than simultaneously solving the two equations of degree 2 and 8.

Proposition 4.1. *Solutions ρ_1 and ρ_2 to the quadratic equation in (3.6) can be parametrized as*

$$\begin{cases} \rho_1(t) = \dfrac{2\alpha t \cos\theta + \beta(1 - t^2)\sin\theta}{1 + t^2} \\ \rho_2(t) = \dfrac{-2\alpha t \sin\theta + \beta(1 - t^2)\cos\theta}{1 + t^2}, \end{cases} \quad -1 \le t \le 1$$

where

$$\tan(2\theta) = \frac{2(1 + \boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_2)}{\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2}, \qquad 0 \le 2\theta \le \pi,$$

and

$$\alpha = \frac{\sqrt{2}\|\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w\|}{\sqrt{(\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2) + (\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2)\cos(2\theta) + 2\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_2 \sin(2\theta)}}, \qquad \alpha > 0,$$

$$\beta = \frac{\sqrt{2}\|\boldsymbol{\Gamma}_1^w - \boldsymbol{\Gamma}_2^w\|}{\sqrt{(\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2) - (\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2^\top \boldsymbol{\gamma}_2)\cos(2\theta) - 2\boldsymbol{\gamma}_1^\top \boldsymbol{\gamma}_2 \sin(2\theta)}}, \qquad \beta > 0.$$

Proof. An ellipse centered at the origin with semi-axes of lengths $\alpha > 0$ and $\beta > 0$ and parallel to the coordinates x and y can be parametrized as

$$x = \frac{2t}{1 + t^2}\alpha, \quad y = \frac{(1 - t^2)}{1 + t^2}\beta, \qquad t \in (-\infty, \infty), \tag{4.1}$$

with ellipse vertices identified at $t = -1$, 0, 1 and ∞, as shown in Fig. 6. For a general ellipse centered at the origin, the coordinates must be multiplied with the rotation matrix for angle θ, obtaining

$$\begin{cases} \rho_1 = \dfrac{2\alpha t \cos\theta + \beta(1 - t^2)\sin\theta}{1 + t^2} \\ \rho_2 = \dfrac{-2\alpha t \sin\theta + \beta(1 - t^2)\cos\theta}{1 + t^2}. \end{cases} \quad -1 \le t \le 1$$

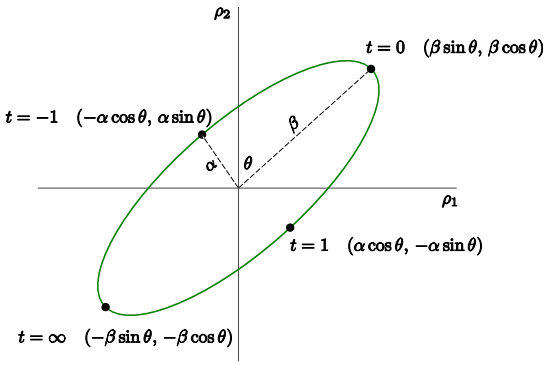

Fig. 6. A parametrization of the ellipse by a parameter t.

Fig. 6 illustrates this parametrization. Notice that the range of values of t we need to consider certainly lies in $[-1, 1]$ and in fact in a smaller interval where $\rho_1 > 0$ and $\rho_2 > 0$. Note that t and $-\frac{1}{t}$ correspond to opposite points on the ellipse.

The parameters α, β, and θ for the ellipse in Eqs. 3.26 and 3.6 can be found by substitution of ρ_1 and ρ_2 in the parametric form into Eq. 3.26. Specifically, writing

$$\begin{aligned}&\frac{\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1}{(1+t^2)^2}[4\alpha^2t^2\cos^2\theta+\beta^2(1-t^2)^2\sin^2\theta+4\alpha\beta t(1-t^2)\sin\theta\cos\theta]\\&\quad-\frac{2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2}{(1+t^2)^2}[-4\alpha^2t^2\sin\theta\cos\theta+2\alpha\beta t(1-t^2)\cos^2\theta\\&\quad-2\alpha\beta t(1-t^2)\sin^2\theta]+\beta^2(1-t^2)^2\sin\theta\cos\theta\\&\quad-\frac{2\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2}{(1+t^2)^2}[4\alpha^2t^2\sin^2\theta+\beta^2(1-t^2)^2\cos^2\theta\\&\quad-4\alpha\beta t(1-t^2)\sin\theta\cos\theta]=\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|^2.\end{aligned} \tag{4.2}$$

Simplifying the equation as

$$\begin{aligned}&[\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1 4\alpha^2t^2-\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 4\alpha\beta t(1-t^2)+(\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)\beta^2(1-t^2)^2]\cos^2\theta\\&+[\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1\beta^2(1-t^2)^2+\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 4\alpha\beta t(1-t^2)\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2 4\alpha^2t^2]\sin^2\theta\\&+[\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1 4\alpha\beta t(1-t^2)+(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 8\alpha^2t^2-\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 2\beta^2(1-t^2)^2\\&-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2 4\ \ \beta t(1-t^2)]\sin\theta\cos\theta=(1+t^2)^2\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|^2\end{aligned} \tag{4.3}$$

and using 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 $\sin(2\theta)=2\sin\theta\cos\theta$,

$$\begin{aligned}&[\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1 4\alpha^2t^2-\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 4\alpha\beta t(1-t^2)+\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2\beta^2(1-t^2)^2]\\&\quad\cdot(1+\cos(2\theta))+\\&[\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1\beta^2(1-t^2)^2+\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 4\alpha\beta t(1-t^2)+\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2 4\alpha^2t^2]\\&\quad\cdot(1-\cos(2\theta))+\\&[\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1 4\alpha\beta t(1-t^2)+\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 8\alpha^2t^2-\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 2\beta^2(1-t^2)^2\\&\quad-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2 4\alpha\beta t(1-t^2)]\sin(2\theta)=2(1+t^2)^2\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|^2.\end{aligned} \tag{4.4}$$

which is an equation only involving the unknown θ,

$$\begin{aligned}&(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1+\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)[4\alpha^2t^2+\beta^2(1-t^2)]+\\&[(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)[4\alpha^2t^2-\beta^2(1-t^2)^2]-\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 8\alpha\beta t(1-t^2)]\cos2\theta\\&[(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)4\alpha\beta t(1-t^2)+2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2[4\alpha^2t^2-\beta^2(1-t^2)^2]]\sin2\theta\\&=2(1+t^2)^2\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|^2.\end{aligned} \tag{4.5}$$

This equation holds for all values of t. For $t=0$,

$$\begin{aligned}&(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1+\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)\beta^2-(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)\beta^2\cos2\theta\\&\quad-2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2\beta^2\sin2\theta=2\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|,\end{aligned} \tag{4.6}$$

giving

$$\beta^2=\frac{2\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|^2}{(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1+\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)-(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)\cos2\theta-2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2\sin2\theta}. \tag{4.7}$$

Similarly, at $t=1$,

$$\begin{aligned}&(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1+\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)4\alpha^2+(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)4\alpha^2\cos2\theta\\&\quad+2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2 4\alpha^2\sin2\theta=8\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|^2,\end{aligned} \tag{4.8}$$

giving

$$\alpha^2=\frac{2\|\boldsymbol{\Gamma}_1^w-\boldsymbol{\Gamma}_2^w\|^2}{(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1+\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)+(\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_1-\boldsymbol{\gamma}_2^\top\boldsymbol{\gamma}_2)\cos2\theta+2\boldsymbol{\gamma}_1^\top\boldsymbol{\gamma}_2\sin2\theta}. \tag{4.9}$$

□

Both equations in (3.6) are symmetric with respect to the origin in the (ρ_1, ρ_2)-plane and the curves will intersect in at most $2\times8=16$ real points, at most 8 of which will be in the positive quadrant, as we in fact require $\rho_1 > 0$ and $\rho_2 > 0$.

The parametrization of the ellipse given in Proposition 4.1 allows us to reduce the two Eqs. 3.6 to a single polynomial equation in t. Substituting for ρ_1, ρ_2 in terms of t into $Q=0$ gives an equation in t for which, in fact, all the denominators are $(1+t^2)^{12}$, so that these can be cleared leaving a polynomial in $\tilde{Q}(t)$ of degree 16. The symmetry with respect to the origin in the (ρ_1, ρ_2)-plane becomes, in terms of t, a symmetry with respect to the substitution $t \to -1/t$, which gives diametrically opposite points of the ellipse. This implies that $\tilde{Q}$ has the special form

$$\tilde{Q}(t)=q_0+q_1t+q_2t^2+\cdots+q_{16}t^{16}, \tag{4.10}$$

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 $\rho_2 > 0$.

5 Experiments

The Matlab code for this paper is available at github.com/rfabbri/diffgeom2pose. Our goal was to provide a reference and reliable implementation rather than highly optimized code. We use two sets of experiments to probe camera pose recovery: synthetic and real data. For the synthetic data, we employ two datasets: a simple random synthetic dataset, and a more extensive, realistic dataset consisting in a variety of analytic 3D curves (helices, parabolas, ellipses, straight lines, and saddle curves), Fig. 7. For the real data we test on two different datasets and compare against the dataset ground-truth. For the synthetic dataset, in addition to comparing against ground truth we compare our Matlab reference implementation against the state-of-the-art P3P solver recently made available in the OpenMVG SfM pipeline [38] as highly optimized C++ code. Experiments were run on a cluster of Intel Xeon E5-2670 with GNU Parallel [39].

5.1 Synthetic experiments

The first experiment is a test with perfect correspondence, exercising a wide range of random data variation to provide a ballpark on robustness and speed of the solver, taking an optimized P3P implementation as reference. Ground-truth translations are normally sampled around $(0, 0, 10)^\top$ with $\sigma=1$, while camera rotations are uniformly sampled; points in space are normally sampled around the origin with $\sigma=1$, and 3D tangents are

uniformly generated, then projected. The pose is estimated by our P2PT algorithm and by P3P from OpenMVG for each sample. Rotation error is measured by the angle (around an axis) of the rotation that aligns the reconstructed camera to the ground truth [40]. For 10^5 repetitions of this initial experiment, the average time is 15 ms for the proposed P2PT in Matlab – less than half if adjusting for smaller number of required RANSAC runs – vs 12 µs for P3P in C++. This is remarkably efficient given that our Matlab implementation has no code optimizations and the problem degree is 4 times higher. The rotation and translation errors had mean at 10^{-13} for P2PT vs 10^{-7} for P3P. The mean number of solutions was 4 for both solvers. We next explore these statistics in more detail, together with stability tests for a more realistically-structured synthetic dataset.

Our main synthetic dataset developed for this paper is publically available at multiview-3d-drawing.sourceforge.net. The analytic space curves shown in Fig. 7 are synthesized in a $4\times4\times4\,\text{cm}^3$ volume and projected to 100 cameras and are sampled to get 5117 potential data points/tangents that are the projections of the same 3D analytic points and tangents. Camera centers are randomly sampled around an average sphere around the scene along normally distributed radii of mean 1 m and $\sigma = 10$ mm. Rotations are constructed via normally distributed look-at directions with mean along the sphere radius looking to the object, and $\sigma = 0.01$ rad such that the scene does not leave the viewport, followed by uniformly distributed roll. This sampling is filtered such that no two cameras are within 15° of each other. Each camera encompasses a 500×400 pixel viewport, where the entire dataset is always visible at sub-pixel precision with no more than one sample per pixel. These curve samples are then degraded with noise and mismatches. The image location and tangent orientation are perturbed to simulate measurement noise in the range of 0-2 pixels in location and 0-10° in orientation. We add uniform noise to each point coordinate in the range $(-\Delta_{\text{pos}}, \Delta_{\text{pos}})$, with $\Delta_{\text{pos}} \in \{0, 0.5, 1, 2\}$, and add uniform angular noise to the tangent vector in the range $(-\Delta_\theta, \Delta_\theta)$ for $\Delta_\theta \in \{0°, 0.5°, 1°, 5°, 10°\}$. Our expectation using the publically available edge detector [20] is that the edges can be found with subpixel accuracy and edge orientations are guaranteed accurate to less than 5° [19].

To simulate the intended application, pairs of 2D-3D point-tangent correspondences for each of the 100 views are selected in a RANSAC procedure from among 5117 veridical ones, to which 50% random spurious correspondences were added. The proposed P2PT solver estimates the pose of the camera inside the RANSAC loop. Only 17 P2PT runs suffice to hit an outlier-free correspondence pair 99% of the time, rather than 35 P3P runs; or only 32 P2PT runs suffice for 99.99% probability rather than 70 P3P runs. For the same number of runs, P3P will fail orders of magnitude more often than P2PT, as shown in the Introduction. To be fair when comparing with P3P, the pose of each of the 100 views in the dataset was estimated using 100 RANSAC iterations for both P2PT and P3P.

To assess the output of the algorithm, we not only measure the error of the estimated pose compared to the ground truth, but, more meaningfuly, the impact of the measured pose on the *measured* reprojection error. Since this is a controlled experiment, we measure final reprojection error not just to the inlier set, but to the entire pool of 5117 true correspondences. In practice, a bundle-adjustment is always run on the results with the greatest number of inliers, to refine the pose estimate using all inliers; but we first report the raw intermediate errors without nonlinear least-squares refinement. The distribution of individual pixel-wise reprojection error of each of the 100 RANSAC solves (having 100 iterations each) is plotted for various levels of measurement noise, Fig. 8(left). What is shown for each Δ_{pos} and Δ_θ is the standard boxplot of all 511,700 (5117 points across the 100 views) reprojection errors. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, resp. The whiskers extend to the most extreme data points not considered as generic outliers (not RANSAC outliers), and outliers are plotted individually using dot marks; an outlier is a value that is more than 1.5 times the interquartile range away from the top or bottom of the box. A small fraction of outliers is expected as long as the cumulative error is not significantly affected.

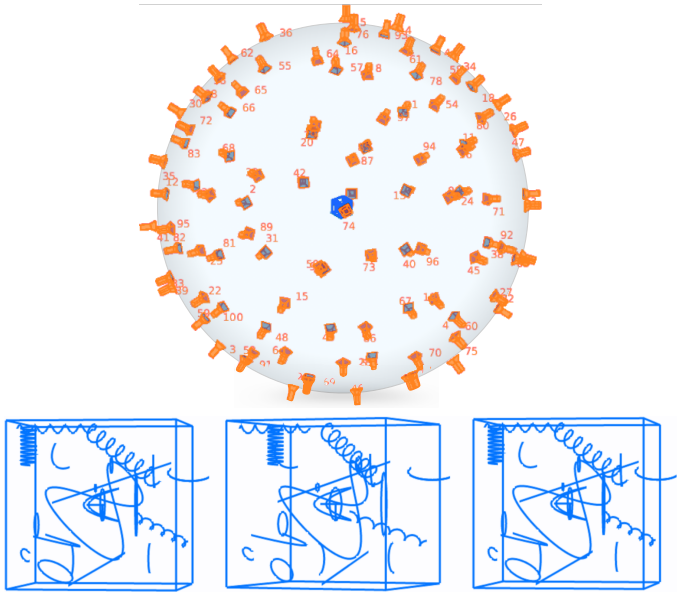

Fig. 7. Our synthetic dataset comprises 100 random views of analytic space curves on a near-spherical configuration (top), at human scale. Sample views and 3D curves are shown in the bottom row for stereo viewing with parallel eyes (left pair) or with crosed eyes (right pair).

The plots show that the relative camera pose can be effectively determined for a viable range of measurement errors, specially since these results are typically optimized in practice through bundle adjustment. We also ran out-of-the-box point-based bundle adjustment on top of the RANSAC results, which is standard practice, and recorded the distribution of reprojection errors, shown in Fig. 8 (right). The plots in Fig. 8 show that both the positional and tangential accuracy of the data are important to constrain the camera pose using the proposed technique. With (point-based) bundle adjustment, the distributions become sharper for each level of localization noise, most with the same median as before bundle adjustment that reflects the synthesized ground-truth noise. For each of these, the tangent noise does not affect the plot anymore, since the bundle adjustment is not using tangent information; more importantly, this shows that all the solutions found using our technique are within the optimal convergence basin for bundle adjustment (optimal in the sense of having the minimum reprojection error we synthesized), even when tangents are pertubed to 10°. Therefore, the proposed technique is stable for practical usage. This is confirmed in the remaining plots for final translational and rotational error, Fig. 8, which show good precision, even when the initial perturbations are high.

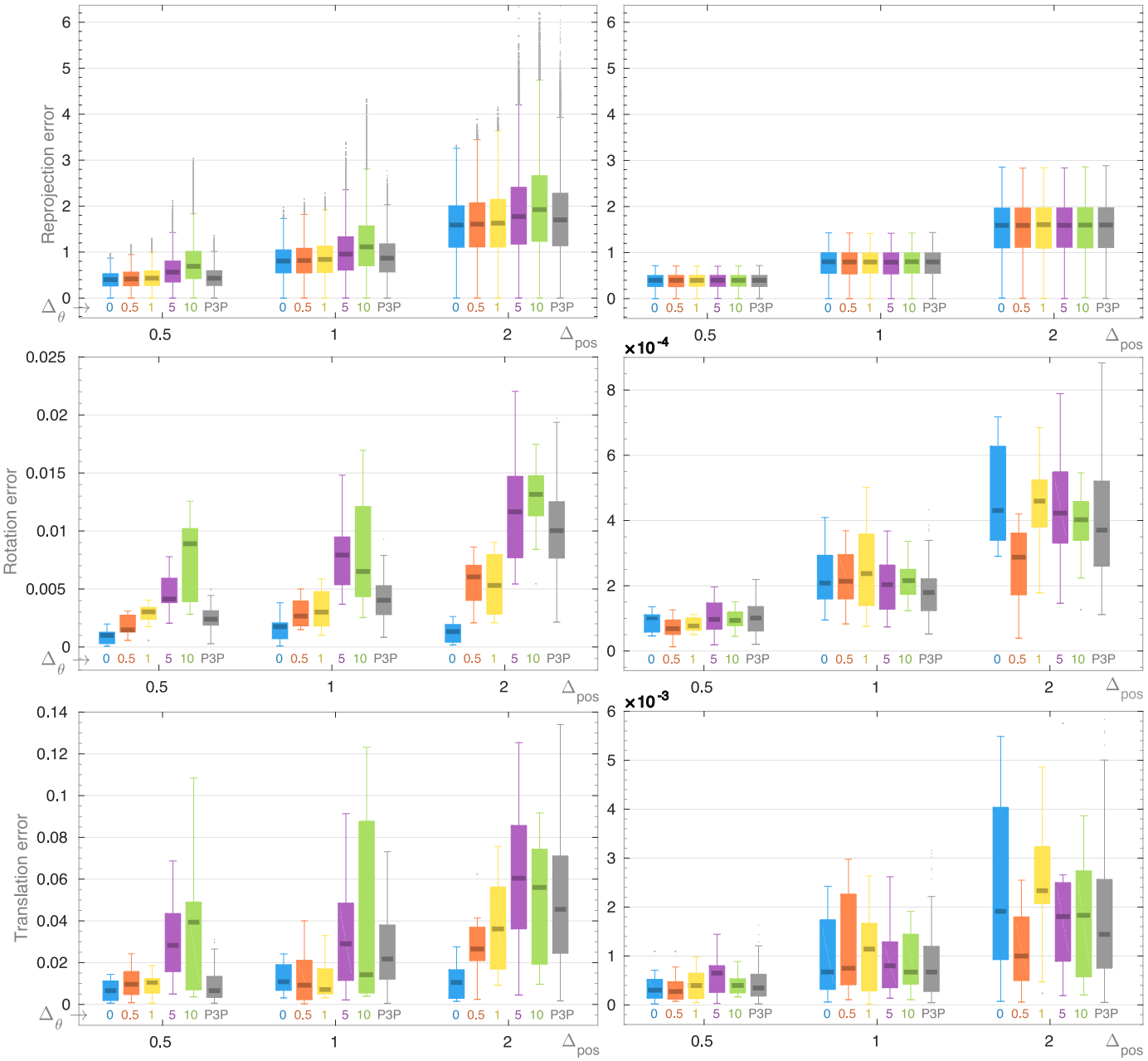

Fig. 8. Distributions of reprojection (top row), rotation (middle), and translation errors (bottom), for using synthetic data of Fig. 7, without bundle adjustment (left column) and after bundle adjustment (right), for levels of positional (x axes) and tangential noise (colors). The proposed P2PT stably matches ground truth and P3P, with only two correspondences. The rotation and translation errors on the right column are close to zero.

5.2 Real data experiments

Capitol sequence. This dataset comprises 256 1280×720 frames covering a 90° helicopter fly-by around the Rhode Island State Capitol, Fig. 2. Intrinsic parameters were initialized using the Matlab toolbox from J. Bouguet. Camera parameters were obtained by running the standard SfM pipeline Bundler [6].

To get point-tangents, we picked a set of manual edge correspondences (in this case 30) across 3 views, and reconstructed a 3D cloud of edges from the first two using the dataset cameras. These first two views are meant to resemble the initial boostrapping of SfM from an Essential matrix technique. The third view plays the role of novel views to be iteratively integrated and registered by an SfM system. The correspondences are obtained by clicking detected image edges [20] on a specialized GUI, using epipolar geometry as a guide. To help determine the ground-truth correspondence, as part of the manual operation within a GUI, in addition to two-view epipolar geometry, we also employ a trinocular stereo constraint based on transfer of differential geometry from two views to a third [12], [22], [41], [42]. The edge detector [20] provides subpixel edges sampled as discrete vector fields (samples of point-tangents) along image curves. We reconstruct the marked 3D edge points using standard two-view triangulation, and their directions using [22, eq. 4.11]. For every new view for which the pose is to be determined, manual 2D edge correspondences to this new view induce a 3D-2D edge correspondence, just as in the Essential matrix + P3P stage of an SfM pipeline. The correspondences across three views give a set of 3D-2D correspondences with which we seek to determine

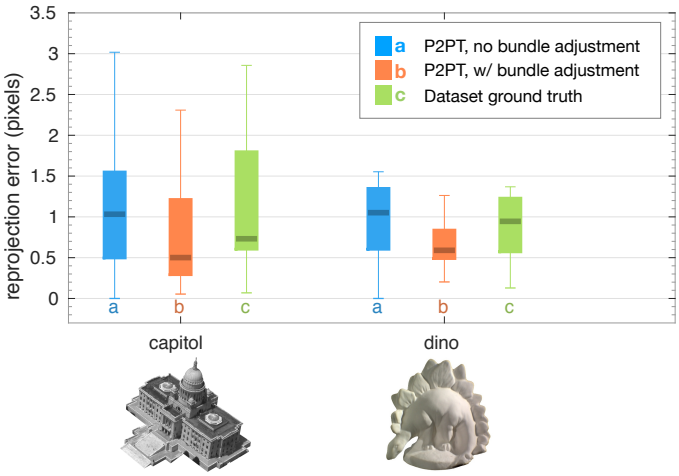

Fig. 9. The reprojection error distribution for real data (Capitol sequence and Middlebury Dino) using only two point-tangents.

the pose of the third view and compare to the dataset pose. Pairs of matches from 3D edges to observed edges in novel views are used with RANSAC to compute the camera pose with respect to the frame of the 3D points, and measure reprojection error. We added 50% outliers to the set of manual correspondences before running RANSAC followed by bundle adjustment.

Fig. 9 shows the error distribution of our method for a single point-tangent pair after RANSAC, before and after running bundle-adjustment, versus the dataset ground-truth from Bundler (which is bundle-adjusted), for the Capitol. P2PT achieved an average error of $1.1px$ and $0.76px$ before and after a metric bundle adjustment, respectively, versus $1.3px$ from Bundler.

Dino sequence. The standard dataset [43], Fig. 9, has 363 views at 640×480 on a hemisphere around the object. The data is low-resolution compared to the Capitol dataset. The ground-truth cameras were generated by estimating the global parameters of a spherical gantry and camera intrinsics with J. Bouguet's toolbox and other software. Laser-scans of the dino were merged and the cameras aligned. Point-tangents were obtained as for the Capitol, but with 10 correspondences. The camera accuracy is 'about' 1-$2px$ according to the authors. Even though this is a strict dataset, the average error using the ground-truth camera is $0.88px$, while the average reprojection error using our method are $1.03px$ and $0.66px$ before and after bundle adjustment, respectively.

6 Conclusions

We presented and characterized a new minimal problem and a practical solver for absolute pose estimation – a P2PT complementary alternative to P3P that requires only two pairs of 3D-2D corresponding points instead of the usual three, where the points are attributed with oriented tangents or incident lines. This is useful for bootstrapping SfM from challenging low-texture data, such as indoor and man-made scenes [8]. It serves as a the core local differential formulation of pose estimation for general curve data when point features are insufficient, but also works in useful particular cases such as corners which have incident lines, widespread in buildings, and in ongoing work we have observed that widespread SIFT feature points with SIFT orientations as the tangents also work well. The proposed theory employs a judicious mixture of algebraic and differential-geometric techniques that exploit the geometry of the problem to produce a stable and relatively fast solver in Matlab, including orientation constraints. We show that this is a 16-degree minimal problem, but careful exploitation of orientation constraints and symmetries drop the number of solutions to fewer than 8. Experiments on real and synthetic data demonstrate the stability and accuracy of the proposed solver, matching the performance of widely used SfM protocols including the OpenMVG P3P implementation. Using one less correspondence drastically reduces RANSAC combinatorics, so P2PT has an advantage over P3P for exceptional levels of robustness and thus potentially in speed, with the caveat that it is solving a not necessarily competing new problem.

We have been actively working on relative pose estimation from corresponding point-tangents across 3 views. Preliminary work [24] has shown that only three triplets of correspondences are sufficient, with fewer if higher-order geometry can be employed. The underlying polynomial is of high degree (320) – an order of magnitude higher than the polynomial treated here. By fusing the detailed approach presented this paper with more generic and scalable techniques from both symbolic and numerical algebraic geometry, we plan to solve a series of related problems to allow a complete curve-based structure from motion system starting from a set of images without any initial pose, with this paper as an alternative to P3P, and the follow-up trifocal method [24] as an alternative to the Essential matrix. We are in the process of integrating these into widely-used C++ SfM pipelines.

Acknowledgments

We acknowledge the NSF grant 1116140, CNPq/Brazil proc. 200875/2004-3, FAPERJ/Brazil E26/112.082/2011, E26/190.180/2010, the 2011 UERJ visiting professor grant, and ICERM NSF grant DMS-1439786 of the 2018 Nonlinear Algebra program and the 2019 Algebraic Vision Research Cluster.

References

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1, 2, 3, 4

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. 1

[3] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981. 1, 3

[4] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm, "Reconstructing the world in six days," in *Computer Vision and Pattern Recognition*, 2015. 1

[5] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004. 1

[6] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 2009. 1, 10

[7] Y. Diskin and V. Asari, "Dense point-cloud representation of a scene using monocular vision," *Journal of Electronic Imaging*, vol. 24, no. 2, p. 023003, 2015. [Online]. Available: http://dx.doi.org/10.1117/1.JEI.24.2.023003 1

[8] A. A. Team, "Understanding arkit tracking and detection," 2018, WWDC. [Online]. Available: https://developer.apple.com/videos/play/wwdc2018/610 1, 11

[9] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *Int. J. Comput. Vision*, vol. 73, no. 3, pp. 263–284, 2007. 1

[10] F. Simoes, M. Almeida, M. Pinheiro, and R. dos Anjos, "Challenges in 3D reconstruction from images for difficult large-scale objects," *2014 XVI Symposium on Virtual and Augmented Reality*, vol. 0, pp. 74–83, 2012. 1

[11] Y. Shinozuka and H. Saito, "Sharing 3D object with multiple clients via networks using vision-based 3D object tracking," in *Proceedings of the 2014 Virtual Reality International Conference*, ser. VRIC '14. New York, NY, USA: ACM, 2014, pp. 34:1–34:4. [Online]. Available: http://doi.acm.org/10.1145/2617841.2620723 1

[12] R. Fabbri and B. 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: IEEE Computer Society Press, 2010. 1, 2, 10

[13] A. Usumezbas, R. Fabbri, and B. B. Kimia, "From multiview image curves to 3D drawings," in *Proceedings of the European Conference in Computer Visiohn*, 2016. 1, 2

[14] ——, "The surfacing of multiview 3d drawings via lofting and occlusion reasoning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 3

[15] R. Fabbri, "Multiview differential geometry in application to computer vision," Ph.D. Dissertation, Division Of Engineering, Brown University, Providence, RI, 02912, July 2010. 1

[16] R. Cipolla and P. Giblin, *Visual Motion of Curves and Surfaces*. Cambridge University Press, 1999. 1, 3

[17] I. Nurutdinova and A. Fitzgibbon, "Towards pointless structure from motion: 3d reconstruction and camera parameters from general 3d curves," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2363–2371. 1

[18] L. Liu, D. Ceylan, C. Lin, W. Wang, and N. J. Mitra, "Image-based reconstruction of wire art," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 63:1–63:11, Jul. 2017. [Online]. Available: http://doi.acm.org/10.1145/3072959.3073682 1

[19] B. B. Kimia, X. Li, Y. Guo, and A. Tamrakar, "Differential geometry in edge detection: Accurate estimation of position, orientation and curvature," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1573–1586, July 2019. 2, 9

[20] A. Tamrakar and B. B. Kimia, "No grouping left behind: From edges to curve fragments," in *Proceedings of the IEEE International Conference on Computer Vision*. Rio de Janeiro, Brazil: IEEE Computer Society, October 2007. 2, 9, 10

[21] Y. Guo, N. Kumar, M. Narayanan, and B. Kimia, "A multi-stage approach to curve extraction," in *CVPR'14*, 2014. 2

[22] R. Fabbri and B. B. Kimia, "Multiview differential geometry of curves," *International Journal of Computer Vision*, vol. 117, pp. 1–23, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11263-016-0912-7 2, 3, 4, 10

[23] ——, "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*. Springer Verlag, November 2005, pp. 645–660. 2

[24] R. Fabbri, T. Duff, H. Fan, M. H. Regan, D. da Costa de Pinho, E. Tsigaridas, C. Wrampler, J. D. Hauenstein, B. B. Kimia, A. Leykin, and T. Pajdla, "Trifocal relative pose from lines at points and its efficient solution," *CoRR*, vol. abs/1903.09755, 2019. [Online]. Available: http://arxiv.org/abs/1903.09755 2, 3, 11

[25] D. Barath, "Five-point fundamental matrix estimation for uncalibrated cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 235–243. 2

[26] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981. 2

[27] J. Porrill and S. Pollard, "Curve matching and stereo calibration," *Image and Vision Computing*, vol. 9, no. 1, pp. 45–50, 1991. 3

[28] F. Kahl and A. Heyden, "Using conic correspondence in two images to estimate the epipolar geometry," in *Proceedings of the IEEE International Conference on Computer Vision*. Bombay, India: IEEE Computer Society Press, 1998, p. 761. 3

[29] J. Y. Kaminski and A. Shashua, "Multiple view geometry of general algebraic curves," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 195–219, February 2004. 3

[30] S. N. Sinha, M. Pollefeys, and L. McMillan, "Camera network calibration from dynamic silhouettes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2004, pp. 195–202. 3

[31] J. A. Grunert, "Das pothenotische problem in erweiterter gestalt nebst Über seine anwendungen in der geodäsie," *Archiv der für Mathematik and Physik*, vol. 1, pp. 238–248, 1841. 3

[32] S. Finsterwalder and W. Scheufele, "Das ruckwartseinschneiden im raum," *Sebastian Finsterwalder zum 75*, pp. 86–100, 1937. 3

[33] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, "An analytic solution for the perspective 4-point problem," *Computer Vision, Graphics, and Image Processing*, vol. 47, no. 1, pp. 33–44, 1989. 3

[34] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *Int. J. Comput. Vision*, vol. 13, no. 3, pp. 331–356, 1994. 3

[35] 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.*, vol. 24, no. 4, pp. 550–555, 2002. 3

[36] M. Bujnak, Z. Kukelova, and T. Pajdla, "A general solution to the p4p problem for camera with unknown focal length," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2008. 3

[37] Y. Kuang and K. Åström, "Pose estimation with unknown focal length using points, directions and lines," in *International Conference on Computer Vision*. IEEE, 2013, pp. 529–536. 3

[38] M. Persson and K. Nordberg, "Lambda twist: An accurate fast robust perspective three point (p3p) solver," in *The European Conference on Computer Vision (ECCV)*, September 2018. 3, 8

[39] O. Tange, *GNU Parallel 2018*. Ole Tange, Mar. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1146014 8

[40] S. Bianco, G. Ciocca, and D. Marelli, "Evaluating the performance of structure from motion pipelines," *Journal of Imaging*, vol. 4, no. 8, 2018. [Online]. Available: https://www.mdpi.com/2313-433X/4/8/98 9

[41] N. Ayache and L. Lustman, "Fast and reliable passive trinocular stereovision," in 1^{st} *International Conference on Computer Vision*, June 1987. 10

[42] L. Robert and O. D. Faugeras, "Curve-based stereo: figural continuity and curvature," in *Proceedings of Computer Vision and Pattern Recognition*, June 1991, pp. 57–62. 10

[43] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2006, pp. 519–528. 11

Ricardo Fabbri is a tenured professor of computational modeling at the Rio de Janeiro State University, Brazil. He holds a Ph.D. from Brown University and have worked at Google in offline indexing from images. He has recently organized the Algebraic Vision Research Cluster at ICERM, on multiple view geometry and differential geometry. He is currently interested in mathematics and diffusion maps for the photogrammetric modeling of the water surface.

Peter J. Giblin is Professor of Mathematics Emeritus at the University of Liverpool, UK. His current research interests are in applications of singularity theory to problems of differential geometry. He also works with local high schools in supervising students in advanced project work in mathematics. In the queens birthday honours 2018 he was awarded an OBE for services to mathematics.

Benjamin B. Kimia is Professor of Engineering at Brown Unviersity and holds a Ph.D. degree in Electrical and Computer Engineering from McGill University. His research includes computer vision and medical imaging inspired by neurophysiology and psychophysics. His expertise includes the representation of shape in 2D, 3D and multiview reconstruction, applied to large image databases, archaeology, assistance for the blind, odometry, and image-guided treatments.