Trifocal Relative Pose from Lines at Points and its Efficient Solution

Expanded Version with Supplementary Material

Ricardo Fabbri
Rio de Janeiro State University
rfabbri@iprj.uerj.br

Timothy Duff
Georgia Tech

Hongyi Fan
Brown University

Margaret Regan
University of Notre Dame

David da Costa de Pinho
UENF – Brazil

Elias Tsigaridas
INRIA Paris

Charles Wampler
University of Notre Dame

Jonathan Hauenstein
University of Notre Dame

Peter J. Giblin
University of Liverpool

Benjamin Kimia
Brown University

Anton Leykin
Georgia Tech

Tomas Pajdla
CTU, Prague

Abstract

We present a new minimal problem for relative pose estimation mixing point features with lines incident at points observed in three views and its efficient homotopy continuation solver. We demonstrate the generality of the approach by analyzing and solving an additional problem with mixed point and line correspondences in three views. The minimal problems include correspondences of (i) three points and one line and (ii) three points and two lines through two of the points which is reported and analyzed here for the first time. These are difficult to solve, as they have 216 and – as shown here – 312 solutions, but cover important practical situations when line and point features appear together, e.g.*, in urban scenes or when observing curves. We demonstrate that even such difficult problems can be solved robustly using a suitable homotopy continuation technique and we provide an implementation optimized for minimal problems that can be integrated into engineering applications. Our simulated and real experiments demonstrate our solvers in the camera geometry computation task in structure from motion. We show that new solvers allow for reconstructing challenging scenes where the standard two-view initialization of structure from motion fails.*

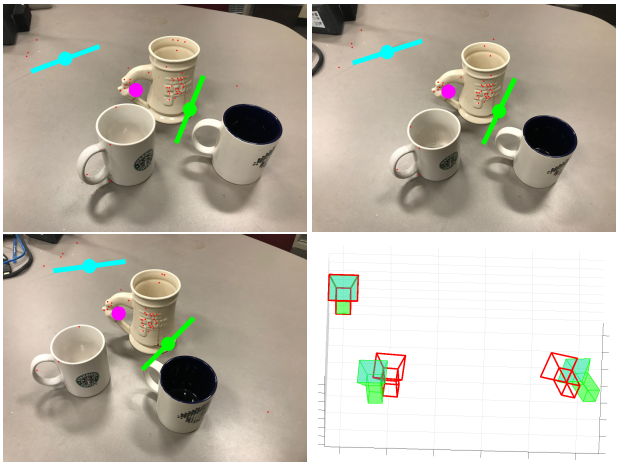

Figure 1. Three mug images illustrate deficiencies of the traditional two-view approach to bootstraping SfM: there is not enough features detected and thus a SOTA SfM pipeline COLMAP [66] fails to reconstruct the relative pose of the cameras. In contrast, the truly trinocular method proposed here used the two triplets of point-tangents and one triplet of points (highlighted among red features) to reconstruct the pose of the cameras as shown. The schematic shows the matching of two triplets of points with attached lines in green and cyan and one point triplet without lines in pink. Red cameras were computed by our solver, and green cameras are the ground truth.

1. Introduction

Three-dimensional computer vision has made a wider impact [4], in part by relying on point-based structure from motion (SfM) [1, 66]. Matching point features across views leads to successful pose estimation and unorganized 3D point cloud reconstructions [52, 21]. Even production-quality SfM technology nevertheless fails [4] when the images contain (i) large homogeneous areas with few or no features; (ii) repeated textures, like brick walls, giving rise to a large number of ambiguously correlated features; (iii) blurred areas, arising from fast moving cameras or objects; (iv) large scale changes where the feature overlap is not sufficiently significant; (v) Multiple and independently moving objects each of which do not have a sufficient number of features.

We track the failure cases to two key observations. First, multiview applications rarely make use of the full information available in the image sequence. Most traditional multiview pipelines estimate the relative pose of two views, say with the 5-point algorithm [50] and then register new views

using a P3P algorithm [67]. Camera estimation from trifocal tensors is long believed to augment two-view pose estimation [19], although this is questioned in practice [31]. The calibrated trinocular relative pose estimation from four points, 3v4p, is known to be difficult to solve [51, 60, 61], partly because it is not a minimal problem since it is overconstrained. The first working solver [51] is effectively parametrizing the relative pose between two cameras in the form of a curve of degree ten of possible epipoles and using a third view to select the one that minimizes reprojection errors. In this sense, trinocular pose estimation has not truly been tackled as a minimal problem.

The second key observation is that low number of point features in images may often be supported by lines and curves. However, the use of points on curves to establish correspondence faces its own challenges. They are only transversally localized, leaving thus a dimension of ambiguity in determining curve correspondence. Despite this, curve points offer additional useful constraint, *i.e. the orientation of their tangent*. Thus, at corners, junctions, and other special points on curves, e.g., satisfying certain appearance conditions (maximizing the cornerness or Laplacian of Gaussian along the curve), enough points are both spatially localized, and orientation also available for an additional constraint. Of course, the availability of orientation is not exclusive to tangents on a curve; for example, we show how the SIFT dominant direction can be used effectively as an orientation attached to a point. We show that the introduction of "orientation attached to a point" can solve for estimation with fewer point matches, from 4 to 3, which is critical in images experiencing a feature drought, Figure 1, as well as to enhance the robustness and speed in RANSAC.

The two above observations motivate exploring trinocular pose estimation from the perspective of triplet point correspondences where the points may also be endowed with orientation. We demonstrate that only three points are needed when matched across all three views: Two of these triplets need to have attached orientation; the third does not; see the schematic in Figure 1.

Three types of constraints arise in matching points with attached orientation. First, the point location correspondence, *i.e.*, the epipolar constraint, provides an equation for each pair of views, or six equations in all. The fact that a pair from view 1 to view 2 and a pair from 2 to view 3 form a triplet provides another equation which essentially constrains the independent pairs of scale ambiguities to a single one. This provides another three equations. Finally, for each triplet of points with attached orientations, the orientation of the first two views predicts an orientation for the third, providing an additional constraint for each triplet with orientation. This provides two more equations, for a total of 11 equations in 11 unknowns.

These equations are polynomial with such complexity that is not trivial to solve efficiently. This motivates using techniques from numerical algebraic geometry [8, 14, 43] to (i) probe whether the system is over or under constrained or otherwise minimal; (ii) understand the range of the number of solutions and a tight upper bound on it; (iii) develop efficient and practically relevant methods for finding solutions which are real and represent camera configurations. This paper answers all three points: the problem posed is minimal, it has up to 312 solutions of which 2-3 end up becoming relevant to camera configurations, and the paper develops a practical and relatively fast method (currently under 2 seconds but promises to be sub-second with some optimization) for solving the system; these are the key contribution of this paper. As a bonus, a similar trifocal problem with three points and a free line is analyzed to demonstrate generality of this approach.

Experiments are conducted on synthetic data to understand how the approach behaves under (i) veridical and accurate correspondences, (ii) veridical but noisy correspondences, and (iii) veridical noisy correspondences embedded among outliers. These experiments demonstrate that the system is robust and stable under spatial and orientation noise and under a significant level of outliers. For experiments on real data, we use SIFT keypoints endowed with SIFT orientation. The approach applies RANSAC to triplets of features, which are essentially pairs that are cycle consistent across three views, and solves the system of polynomial equations using an efficient implementation of homotopy continuation. The results are validated by measuring inliers. We have found that our approach is successful in all cases where the traditional SfM pipeline succeeds but more importantly it succeeds in many other cases too, on the EPFL [70] and Amsterdam Teahouse datasets [71], Figures 1 and 17. For additional details, we refer the reader to the supplementary material.

1.1. Literature Review

Trifocal Geometry Calibrated trifocal geometry estimation is a hard problem [60, 61, 51, 63]. There are no publicly available solvers we are aware of. The state of the art solver [51], based on four corresponding points (3v4p), has not yet found many practical applications [38].

For the uncalibrated case, 6 points are needed [24], and Larsson et al. recently solved the longstanding trifocal minimal problem of using 9 lines [39]. The case of mixed points and lines is less common [54, 54], but has seen a growing interest in related problems [72, 59]. The calibrated cases beyond 3v4p are largely unsolved, spurring more sophisticated theoretical work [33, 42, 45, 46, 2, 53, 3]. Kileel [33] studied many minimal problems in this setting, such as the Cleveland problem solved in the present paper, and reported studies using homotopy continuation. Kileel also stated that

the *full* set of ideal generators is currently unknown, *i.e.*, a given set of polynomial equations provably necessary and sufficient to describe calibrated trifocal geometry.

Seminal works used curves and edges in three views to transfer differential geometry for matching [5, 62], and for pose and trifocal tensor estimation [13, 65]. Point-tangents can be framed as *quivers* (1-quivers), or feature points with attributed directions (*e.g.*, corners), proposed in the context of uncalibrated trifocal geometry but de-emphasizing the connection to tangents to general curves [29, 74]. We note that point-tangent fields may also be framed as vector fields, so related technology may apply to surface-induced correspondence data [16]. In the calibrated setting, point-tangents were first used for absolute pose estimation by Fabbri *et al*. [17], using only two points, later relaxed for unknown focal length [37]. The trifocal problem with three point-tangents as a local version of trifocal pose for global curves was first formulated by Fabbri [16], for which we here present a minimal version codenamed Chicago.

Homotopy Continuation The basic theory of Polynomial Homotopy Continuation (HC) [8, 48, 68] was developed in 1976, and guarantees algorithms that are *globally* convergent with probability one from given start solutions. A number of general-purpose HC software have considerably evolved over the past decade [7, 10, 43, 73]. The computer vision community has used HC most notably in the nineties for 3D vision of curves and surfaces for tasks such as computing 3D line drawings from surface intersections, finding the stable singularities of a 3D line drawing under projections, computing occluding contours, stable poses, hidden line removal by continuation from singularitities, aspect graphs, self-calibration, pose estimation [36, 56, 35, 56, 36, 55, 27, 9, 26, 47, 44, 20, 25, 58], as well as for MRFs [49, 9], and in more recent work [23, 15, 64]. An implementation of the early continuation solver of Kriegman and Ponce [35] by Pollefeys is still widely available for low degree systems [57].

As an early example [25], HC was used to find an early bound of 600 solutions to trifocal pose with 6 lines. In the vision community HC is mostly used as an offline tool to carry out studies of a problem before crafting a symbolic solver. Kasten *et al*. [32] recently compare a general purpose HC solver [73] against their symbolic solver. However, their problem is one order of magnitude lower degree than the ones presented here, and the HC technique chosen for our solver [14] is more specific than their use of polyhedral homotopy, in the sense that fewer paths are tracked (*c.f*. the start system hierarchy in [68]).

2. Two Trifocal Minimal Problems

2.1. Basic Equations

Our notation follows [22] with explicit projective scales. A more elaborate notation [13, 17] can be used to express the equations in terms of tangents to curves.

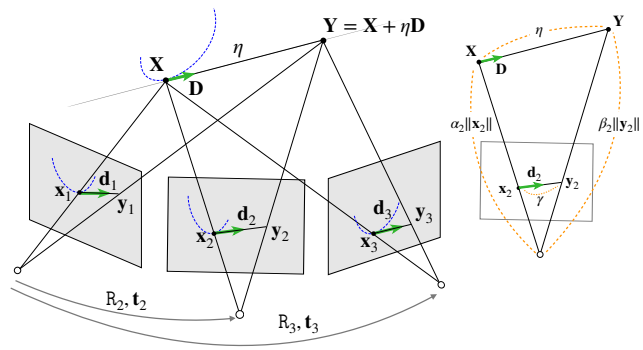

Figure 2. Notation for the trifocal pose problems.

Notation Let $\mathbf{X}$ and $\mathbf{Y}$ denote inhomogeneous coordinates of 3D points and $\mathbf{x}_{v,p} \in \mathbb{P}^2$ denote homogeneous coordinates of image points. Subscript v numbers views and p numbers the points. If only a single subscript is used, it indexes views. Symbols $\mathtt{R}_i, \mathbf{t}_i$ denote the rotation and translation transforming coordinates from camera 1 to camera i, $\mathbf{d}$ is an image line direction or curve tangent in homogeneous coordinates, and $\mathbf{D}$ is the 3D line direction or space curve tangent in inhomogeneous world coordinates. Symbols α, β denote the depth of $\mathbf{X}, \mathbf{Y}$, respectively, and η is the displacement along $\mathbf{D}$ corresponding to the displacement γ_i along $\mathbf{d}$.

We next formulate two minimal problems for points and lines in three views and derive their general equations before turning to specific formulations. We first state a new minimal problem codenamed 'Chicago', followed by an important similar problem, 'Cleveland'.

Definition 1 (Chicago trifocal problem)**.** Given three points $\mathbf{x}_{1v}, \mathbf{x}_{2v}, \mathbf{x}_{3v}$ and two lines ℓ_{1v}, ℓ_{1v} in views $v = 1, 2, 3$, such that the ℓ_{iv} meet $\mathbf{x}_{iv}$, $i = 1, 2$, $v = 1, 2, 3$, compute $\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3$.

Definition 2 (Cleveland trifocal problem)**.** Given three points $\mathbf{x}_{1v}, \mathbf{x}_{2v}, \mathbf{x}_{3v}$ in views $v = 1, 2, 3$, and given one line ℓ_{1v} in each image, compute $\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3$.

To setup equations, we start with image projections of points $\alpha_1\mathbf{x}_1 = \mathbf{X}$, $\alpha_2\mathbf{x}_2 = \mathtt{R}_2\mathbf{X} + \mathbf{t}_2$, $\alpha_3\mathbf{x}_3 = \mathtt{R}_3\mathbf{X} + \mathbf{t}_3$ and eliminate $\mathbf{X}$ to get

$$\alpha_v\mathbf{x}_v = \mathtt{R}_v\alpha_1\mathbf{x}_1 + \mathbf{t}_v, \quad v = 1, 2 \tag{1}$$

Lines in space through $\mathbf{X}$ are modeled by their points $\mathbf{Y} = \mathbf{X} + \eta\mathbf{D}$ in direction $\mathbf{D}$ from $\mathbf{X}$. Points $\mathbf{Y}$ are projected to

images as $\beta_1\mathbf{y}_1 = \mathbf{X} + \eta\mathbf{D}$, $\beta_2\mathbf{y}_2 = \mathtt{R}_2(\mathbf{X} + \eta\mathbf{D}) + \mathbf{t}_2$, $\beta_3\mathbf{y}_3 = \mathtt{R}_3(\mathbf{X} + \eta\mathbf{D}) + \mathbf{t}_3$. Eliminating $\mathbf{X}$ gives

$$\begin{aligned} \beta_1\mathbf{y}_1 &= \alpha_1\mathbf{x}_1 + \eta\mathbf{D} \\ \beta_2\mathbf{y}_2 &= \alpha_2\mathbf{x}_2 + \eta\mathtt{R}_2\mathbf{D} \\ \beta_3\mathbf{y}_3 &= \alpha_3\mathbf{x}_3 + \eta\mathtt{R}_3\mathbf{D} \end{aligned} \tag{2}$$

The directions $\mathbf{d}_i$ of lines in images, which are obtained as the projection of $\mathbf{Y}$ minus that of $\mathbf{X}$, *i.e.*

$$\beta_i\gamma_i\mathbf{d}_i = \mathbf{y}_i - \mathbf{x}_i = \alpha_i\mathbf{x}_i + \eta\mathbf{D} - \mathbf{x}_i, \tag{3}$$

are substituted to (2). After eliminating $\mathbf{D}$ we get

$$(\beta_v - \alpha_v)\mathbf{x}_v + \beta_v\gamma_v\mathbf{d}_v = \mathtt{R}_v\left((\beta_1 - \alpha_1)\mathbf{x}_1 + \beta_1\gamma_1\mathbf{d}_1\right), \tag{4}$$

For $v = 1, 2$. To simplify notation further, we change variables as $\epsilon_i = \beta_i - \alpha_i$, $\mu_i = \beta_i\gamma_i$ and get

$$\epsilon_v\mathbf{x}_v + \mu_v\mathbf{d}_v = \mathtt{R}_v\left(\epsilon_1\mathbf{x}_1 + \mu_1\mathbf{d}_1\right), \tag{5}$$

for $v = 1, 2$. For Chicago, we have three times the point equations (1) and two times the tangent equations (5). There are 12 unknowns $\mathtt{R}_2, \mathbf{t}_2, \mathtt{R}_3, \mathbf{t}_3$, and 24 unknowns $\alpha_{pv}, \epsilon_{pv}, \mu_{pv}$.

For Cleveland we need to represent a free 3D line L in space. We write a general point of L as $\mathbf{P} + \lambda\mathbf{V}$, with a point $\mathbf{P}$ on L, the direction $\mathbf{V}$ of L and real λ. Considering a triplet of corresponding lines represented by their homogeneous coordinates ℓ_v, the homogeneous coordinates of the back-projected planes are obtained as $\pi_v = [\mathtt{R}_v \mid \mathbf{t}_v]^T \ell_v$. Now, all π_v have to contain $\mathbf{P}$ and $\mathbf{V}$ and thus

$$\operatorname{rank}\left[[I \mid 0]^T \ell_v \mid [\mathtt{R}_2 \mid \mathbf{t}_2]^T \ell_v \mid [\mathtt{R}_3 \mid \mathbf{t}_3]^T \ell_v\right] < 3 \tag{6}$$

Equations 6 and 1 are basic equations of Cleveland.

Many ways how to proceed by elimination from basic equations of the problems are possible. A particular formulation based on vanishing minors for both Chicago and Cleveland, which produced our first working solver to Chicago, is described in 3.1.

2.2. Problem Analysis

A general camera pose problem is defined by a list of labeled features in each image, which are in correspondence. The image coordinates of each feature are given, and we are to determine the relative poses of the cameras. The concatenated list of all the features' coordinates from all cameras is a point in the image space Y, while the concatenated list of the features' locations in the world frame or camera 1 is a point in the world feature space W. Unless the scale of some feature is given, the scale of the relative translations is indeterminate, so relative translations are treated as a projective space. For N cameras, the combined poses of cameras $2, \ldots, N$ relative to camera 1 are a point in $SE(3)^{N-1}$. Let the pose space be X, the projectivized version of $SE(3)^{N-1}$, and so $\dim X = 6N - 7$. Given the 3D features and the camera poses, we can compute the image coordinates of the features, so we have a viewing map $V\colon W \times X \to Y$. A camera pose problem is: given $y \in Y$, find $(w, x) \in W \times X$ such that $V(w, x) = y$. The projection $\pi\colon (w, x) \mapsto x$ is the set of relative poses we seek.

Definition 3. A camera pose problem is minimal if $V : W \times X \to Y$ is invertible and nonsingular at a generic $y \in Y$.

A necessary condition for a map to be invertible and nonsingular is that the dimensions of its domain and range must be equal. Let us consider three kinds of features: a point, a point on a line (equivalently a point with tangent direction), and a free line (a line with no distinguished point on it). For each feature, say F, let C_F be the number of cameras that see it. The contributions to $\dim W$ and $\dim Y$ of each kind of feature are in the table below, where a point with a tangent counts as one point and one tangent. Thus, a point feature has several tangents if several lines intersect at it (sometimes called quiver).

Feature	$\dim W$	$\dim Y$
Point, P	3	$2 \cdot C_P$
Tangent, T	2	$1 \cdot C_T$
Free Line, L	4	$2 \cdot C_L$

Accordingly, summing the contributions to $\dim Y - \dim W$ for all the features, we have the following result.

Theorem 2.1. *Let $\langle x \rangle \doteq \max(0, x)$. A necessary condition for a N-camera pose problem to be minimal is*

$$\sum_P \langle 2C_P - 3 \rangle + \sum_T \langle C_T - 2 \rangle + \sum_L \langle 2C_L - 4 \rangle = 6N - 7. \tag{7}$$

For trifocal problems where all cameras see all features, i.e., $C_P = C_T = C_L = 3$, a pose problem with 3 feature points and 2 tangents meets condition (7). A pose problem with 3 feature points and 1 free line also meets the condition. Adding any new features to these problems will make them overconstrained, having $\dim Y > \dim W \times X$.

To demonstrate sufficiency, it enough to find $(w, x) \in W \times X$ where the Jacobian of $V(w, x)$ is full rank. Choosing a random point (w, x) and testing the Jacobian rank serves to establish nonsingularity with probability one. Such a test computed in floating point arithmetic is highly indicative but not rigorous unless one bounds floating-point error, which can be done using interval arithmetic, or exact arithmetic. A singular value decomposition of the Jacobian computed in floating point that shows that the Jacobian has a smallest singular value far from zero, can be taken as a numerical demonstration that the problem is minimal. Similarly, a careful calculation using techniques from numerical algebraic geometry can compute a full solution list in $\mathbb{C}$ for

a randomly selected example and thereby produce a numerical demonstration of the algebraic degree of the problem. Using such techniques, we make the following claims with the caveat that they have been demonstrated numerically, not proven rigorously.

Theorem 2.2 (Numerical)**.** *The Chicago trifocal problem is minimal with algebraic degree 312, and the Cleveland problem is minimal with algebraic degree 216.*

Proof. The previous paragraphs explain the numerical arguments involved, but the definite proof by computer involves symbolically computing the Gröbner basis over $\mathbb{Q}$, with special provisions, as discussed in supplementary material. □

While this result is in agreement with degree counts for Cleveland in [33], the analysis of Chicago is novel as this problem is presented in this paper for the first time. See the supplementary material.

3. Homotopy Continuation Solver

In this section we describe our homotopy continuation solvers. In subsection 3.1 we reformulate the trifocal pose estimation problems as parametric polynomial systems in unknowns $\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3$ using the main specific equations that so far have produced our best results, while other formulations are discussed in suplementary material. We attribute relatively good run times to two factors. First, we use coefficient-parameter homotopy, outlined in 3.2, which naturally exploits the algebraic degree of the problem. Already with general-purpose software [7, 43], parameter homotopies are observed to solve the problems in a relatively efficient manner. Secondly, we optimize various aspects of the homotopy continuation routine, such as polynomial evaluation and numerical linear algebra. In subsection 3.3, we describe our optimized implementation in C++ which was used to do the computations.

3.1. Equations based on minors

One way of building a parametric homotopy continuation solver is to formulate the problems as follows. An instance of Chicago may be described by 5 *visible lines* in each view. We represent each line by its defining equation in homogeneous coordinates, *i.e.* as $\ell_{i,1}, \ldots \ell_{i,5} \in \mathbb{C}^{3\times 1}$ for each $i \in \{1, 2, 3\}$. With the convention that the first three lines pass through the three pairs of points in each view and that the last two pass through associated point-tangent pairs, let

$$L_j = \left[[I \,|0]^T \, \ell_{1,j} \quad [\mathtt{R}_2 \,|\mathbf{t}_2]^T \, \ell_{2,j} \quad [\mathtt{R}_3 \,|\mathbf{t}_3]^T \, \ell_{3,j} \right] \tag{8}$$

for each $j \in \{1, \ldots, 5\}$. We enforce *line correspondences* by setting all 3×3 minors of each L_j equal to zero. Certain *common point constraints* must also be satisfied,*i.e.*, that the

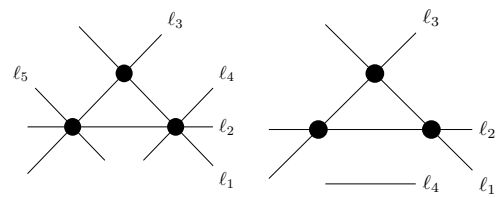

Figure 3. Visible line diagrams for Chicago and Cleveland.

4×4 minors of matrices $[L_1 \mid L_2 \mid L_4]$, $[L_1 \mid L_3 \mid L_5]$, and $[L_1 \mid L_2]$ all vanish.

We may describe the Cleveland problem with similar equations. For this problem, we are given lines $\ell_{i,1}, \ldots, \ell_{i,4}$ for $i \in \{1, 2, 3\}$. We enforce line correspondences for matrices $L_1, \ldots, L_4$ defined as in (8) and common point constraints by requiring that the 4×4 minors of $[L_1 \mid L_2]$, $[L_1 \mid L_3]$, and $[L_2 \mid L_3]$ all vanish. The "visible lines" representation of both problems is depicted in Figure 3.1.

3.2. Algorithm

From the previous section, we may define a specific system of polynomials $F(\mathcal{R}; \mathcal{A})$ in the unknowns $\mathcal{R} = (\mathtt{R}_2, \mathbf{t}_2, \mathtt{R}_3, \mathbf{t}_3)$ parametrized by $\mathcal{A} = (\ell_{1,1}, \ldots)$. Many representations for rotations were explored, but our main implementation employs quaternions. A fundamental technique for solving such systems, fully described in [68], is *coefficient-parameter homotopy.* Algorithm 1 summarizes homotopy continuation from a known set of solutions for given parameter values to compute a set of solutions for the desired parameter values. It assumes that solutions for some starting parameters $\mathcal{A}^*$ have already been computed via some offline, *ab initio* phase. For our problems of interest, the number of start solutions is precisely the algebraic degree of the problem.

Several techniques exist for the *ab initio solve.* For example, one can use standard homotopy continuation to solve the system $F(\mathcal{R}; \mathcal{A}^*) = 0$, where $\mathcal{A}^*$ are randomly generated start parameters [68, 8]. This method may be enhanced by exploiting additional structure in the equations or using regeneration. Another technique based on monodromy, described in [14], was used to obtain a set of starting solutions and parameters for the solver described in Section 3.3.

3.3. Implementation

We provide an optimized C++ package called MINUS – MInimal problem NUmerical Solver `http://github.com/rfabbri/minus`. This is continuation code specialized for minimal problems, templated in C++, so that efficient specialization for different problems and different formulations are made possible. The most reliable and high-quality solver according to our experiments uses a 14×14 minors-based formulation. Although other formulations have demonstrated further potential for speedup by orders

Algorithm 1: Homotopy continuation solution tracker

input: Polynomial system $F(\mathcal{R}; \mathcal{A})$, where
$\mathcal{R} = (\mathtt{R}_2, \mathbf{t}_2, \mathtt{R}_3, \mathbf{t}_3)$, and $\mathcal{A}$ parametrizes the data;
Start parameters $\mathcal{A}^*$; start solutions $\mathcal{R}^*$ where
$F(\mathcal{R}^*; \mathcal{A}^*) = 0$; Target parameters $\widehat{\mathcal{A}}$
output: Set of target solutions $\widehat{\mathcal{R}}$ where $F(\widehat{\mathcal{R}}; \widehat{\mathcal{A}}) = 0$

Setup homotopy $H(\mathcal{R}; s) = F(\mathcal{R}; (1-s)\mathcal{A}^* + s\widehat{\mathcal{A}})$.
for *each start solution* **do**
  $s \longleftarrow \emptyset$
  while $s < 1$ **do**
    Select step size $\Delta s \in (0, 1-s]$.
    Predict: Runge-Kutta Step from s to $s + \Delta s$ such that $dH/ds = 0$.
    Correct: Newton step st. $H(\mathcal{R}; s + \Delta s) = 0$.
    $s \longleftarrow s + \Delta s$

return Computed solutions $\widehat{\mathcal{R}}$ where $H(\widehat{\mathcal{R}}, 1) = 0$.

of magnitude, there may be reliability tradeoffs ($c.f.$ supplementary material).

4. Experiments

We first study the quality of our solver in synthetic experiments. Then, we demonstrate its performance on challenging real data. Due to space constraints, we present results for the Chicago problem, which is more challenging than Cleveland. See the supplementary material for experiments on Cleveland.

Synthetic experiments We show the performance of our solvers by starting with perfect synthetic data [18], consisting of 3D curves in a $4 \times 4 \times 4cm^3$ volume projected to 100 cameras Fig. 4, and sampling them to get 5117 potential data points/tangents that are projections of the same 3D analytic points and tangents [18], and then degrading them with noise and mismatches. Camera centers are randomly sampled around an average sphere around the scene along normally distributed radii of mean 1m and $\sigma = 10mm$, and rotations constructed via normally distributed look-at directions with mean along the sphere radius looking to the object, and $\sigma = 0.01\,\mathrm{rad}$ 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. Our first experiment studies the numerical stability of the solvers. The dataset provides true point correspondences, which inherit an orientation from the tangent to the analytic curve. For each sample set, three triplets of point correspondences are randomly selected with two endowed with the orientation of the tangent to the curve. The real solutions are selected from among the output, and only those that generate positive depth are retained. Finally, the unused tangent of the third triplet is used to verify the solution as it is an overconstrained problem. For each of the remaining solutions a pose is determined.

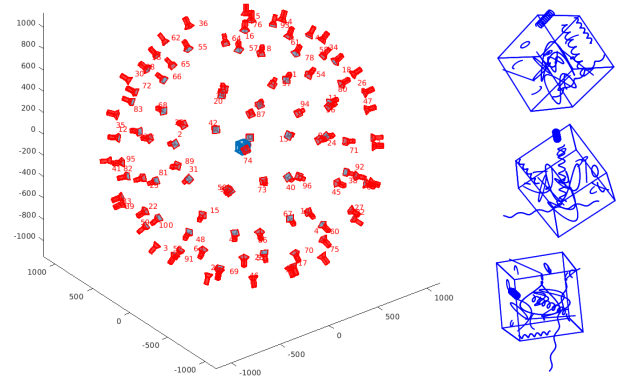

Figure 4. Sample views of our synthetic dataset. Real datasets have also been used in our experiments. (3D curves are from [17, 18]).

The error in pose estimation is compared with the ground truth, measured as the angular error between normalized translation vectors and the angular error between the quaternion representation of the rotation matrices. The entire process of generating the input to computing pose is repeated 1000 times and averaged. This experiment demonstrates that: (i) pose estimation errors are negligible, Fig. 5(a); (ii) the number of solutions is small – 35 real solutions on average which then get pruned down to around 7 on average by enforcing positive depth. Using the unused tangent of the third point as a verification reduces the number of physically realizable solutions to about 3 or 4, Fig. 5(b); (iii) The solver fails in about 1% of cases. These cases are detectable and while not a problem for RANSAC, the solver can be rerun for that solution path with higher accuracy or more parameters at a higher computational cost.

The second experiment shows that we can reliably and accurately determine cameras pose with correct but noisy correspondences. Using the same dataset and a subset of the selection of three triplets of points and tangents – 200 in total – zero-mean Gaussian noise was added both to the feature locations with σ corresponding to $\{0.25, 0.5, 0.75, 1.0\}$ pixels in image and to the orientation of the tangents with $\sigma \in \{0.25, 0.5, 0.75, 1.0\}$. These selected magnitude of localization errors reflect the expected localization error of point features and the orientation error corresponds to the state of the art orientation measurements [34]. A RANSAC scheme determines the feature set with pose generating the highest number of inliers. The experiments indicate that the resulting translation and rotation errors are reasonable. Figure 6(top) shows how changes in the magnitude of feature localization error affect pose in terms of translation errors and rotation errors. We use orientation perturbation of 0.1 rad to simulate the error in real feature orientation. Figure 6(bottom) shows how the magnitude of orientation error affects pose in terms of translation errors and rotation errors.

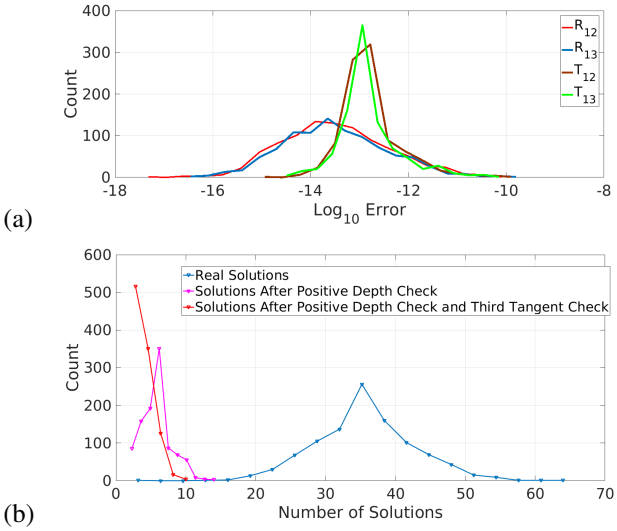

Figure 5. (a) Errors of computed parameters w.r.t the GT are small showing that the solver is numerically stable. (b) The histogram of the numbers of real solutions in different stages.

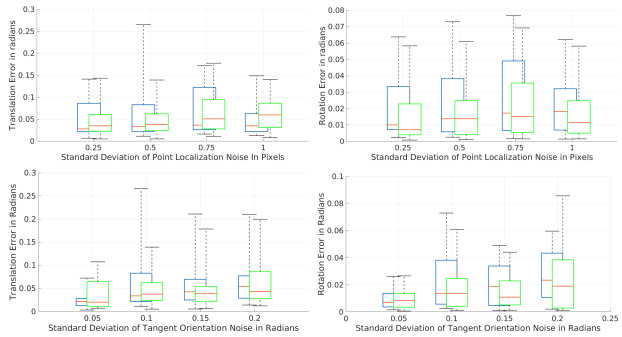

Figure 6. Distribution of trifocal pose error in the form of translational and rotational error is plotted against the level of feature localization noise and orientation noise. The green, resp., blue plots refer to the pose of the second resp., the third camera, relative to the first.

A localization error of 0.5 pixel is used as orientation error is varied.

More meaningful, however, is the error measured in observation space, *i.e.*, the reprojection error: in each triplet of features, the first two features are used to predict the location of the third and the distance between the reprojected feature and the third perturbed feature is the reprojection error. This process is repeated 100 times to generate Figure 7.

The third experiment probes whether the system can reliably and accurately determine trifocal pose when correct noisy correspondences are mixed with outliers. With an error of 0.25 pixels and 0.1 radians, 200 triplets of features were first generated and a percentage of these replaced with samples with random location and orientation. The ratio of outliers is 10%, 25% and 40%. The experiment was repeated 100 times. The resulting reprojection error is low

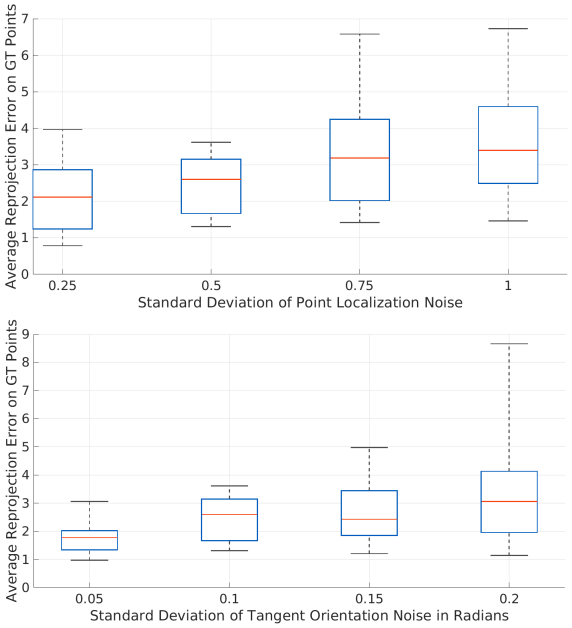

Figure 7. The distribution of reprojection error of feature location is plotted against of levels of feature localization error and feature orientation error.

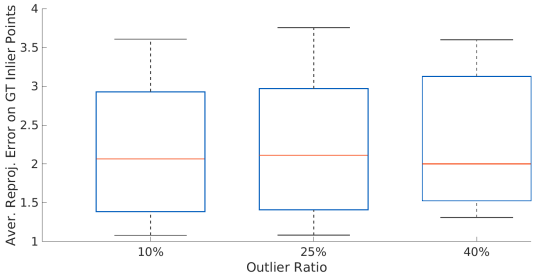

Figure 8. Average reprojection error on GT inlier points with different ratio of outliers.

and stable with the outlier ratio, Fig. 8.

Computational efficiency: Each step of minimal solve using our solver MINUS takes $1.9s$ in the worst case (about $660ms$ on average), corresponding to over 1 minute in our best prototypes using general purpose software [43, 7], both on an Intel core i7-7920HQ processor and four threads. More aggressive but potentially unsafe optimizations towards microseconds are feasible, but require assessing failure rate, as reported in the supplementary materials.

Real experiments: The use of attended lines in our approach requires harvesting points with attached tangents or orientations. In the case of isolated points, such as SIFT keypoints, the orientation of the SIFT descriptor allows a point to be endowed with an orientation. In the case of curves, the curve tangent provides a natural orientation for each point. However, while curve points show superior transversal localization and superior orientation

specification there is correspondence ambiguity along the curve. This can be resolved by employing corners and junctions [28] or special appearance-based keypoints found along a curve. One can also use the curve-to-curve correspondence ambiguity as part of a RANSAC procedure with some help from recent work [41]. These options are all viable. Since the focus of this paper is on the introduction of the approach, the solver, and a practical pipeline for trifocal pose estimation, we focus on the use of SIFT keypoints with SIFT orientations. We recognize that this is suboptimal, as the main drawback of feature-based relative pose estimation is in areas of low-number of features and repeated texture, so working with feature points inherits these difficulties. It would have been better to work with curves which are prominent and stable. Nevertheless we can use SIFT keypoints with attached SIFT orientations to illustrate that our method is at least as good as the traditional methods in all cases and in some cases solves the relative pose when the traditional scheme fails. It is worth emphasizing, however, that the potential of this scheme is to go beyond isolated features, a subject of future work.

Much like the standard pipeline, SIFT features are first extracted from all images. Pairwise features are found by rank-ordering measured similarities and making sure each feature's match in another image is not ambiguous and is above accepted similarity. Pairs of features from the first and second views are then grouped with the pairs of features from the second and third views into triplets. A cycle consistency check enforces that the triplets must also support a pair from the first and third views. Three feature triplets are then selected using RANSAC and together with their assigned SIFT orientation at two points used to estimate the relative pose of the three cameras.

Examples of this procedure are reported for triplets of images taken from the EPFL dense multi-view stereo test image dataset [69] in Figure 17, with ground-truth cameras shown in solid green and the cameras obtained with our method in red outlines. A qualitative visual comparison shows that our estimates are excellent. Quantitively, our estimates have pose errors of 1.5×10^{-3} radians in translation and 3.24×10^{-4} radians in rotation. The average reprojection error is 0.310 pixels. These are comparable or better than the trifocal relative pose estimation methods reported in [30]. Our conclusion for this dataset is that our method is at least as good and often better than the traditional methods. See supplementary data for more examples and a substantiation of this claim.

The EPFL dataset, however, is texture-rich, typically yielding on the order of 1000 triplet features per triplet of images. As such it does not portray the typical problems faced in the really challenging situations when there are few features available or when there are repeated textures. The Amsterdam Teahouse Dataset [71], which also has ground-truth relative pose data, depicts scenes with fewer features. Figure 10 shows a triplet of images from this dataset where there is sufficient set of features (the soup can) to support a bifocal relative pose estimation followed by a P3P registration to a third view (using COLMAP [66].) However, when the number of features is reduced, as in Figure 10 where the number of features is much lower (soup can is invisible), COLMAP fails to find relative pose between pairs of these images. In contrast, our approach which relies on three and not five features is able to operate on this scene and recover the camera pose. Figure 1 shows another example. Further results are shown in supplementary material.

Figure 9. Trifocal relative pose estimation of EPFL dataset. For each row, image triplets samples are shown. The estimation results are shown on the right. Ground truth poses are in solid green and estimated poses are in red. More examples in supplementary material.

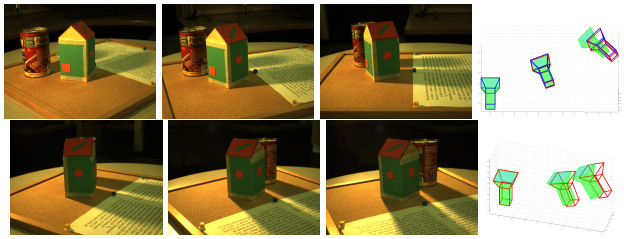

Figure 10. Two samples of trifocal relative camera pose estimation of Amsterdam Teahouse dataset. First line is a sample triplet of images that COLMAP is able to tackle. Second line is a sample triplet from the images that COLMAP reported it cannot find good matches. COLMAP results are in blue wireframes.

5. Conclusion

We presented a new calibrated trifocal minimal problem, an analysis demonstrating its number of solutions, and a practical solver by specializing numerical algebraic computation techniques. We show these techniques generalize to another difficult minimal problem with mixed points and lines. The proposed problem connects classical multi-view geometry of points and lines to that of points and tangents appearing when observing 3D curves extracted with tools of differential geometry [18, 16]. We believe that our approach to solving minimal problems may be useful for other

difficult minimal problems. In the future, our "100 lines of custom-made solution tracking code" will be used to try to improve solvers of many other minimal problems which could not have been solved efficiently with Gröbner basis techniques [40].

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day.

[2] C. Aholt and L. Oeding. The ideal of the trifocal variety. *Math. Comp.*, 83, 2014.

[3] A. Alzati and A. Tortora. A geometric approach to the trifocal tensor. *Journal of Mathematical Imaging and Vision*, 38(3):159–170, Nov 2010.

[4] ARKit Team. Understanding ARKit tracking and detection. Apple, WWDC, 2018.

[5] N. Ayache and L. Lustman. Fast and reliable passive trinocular stereovision. In 1st *International Conference on Computer Vision*, June 1987.

[6] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler. Adaptive multiprecision path tracking. *SIAM Journal on Numerical Analysis*, 46(2):722–746, 2008.

[7] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Bertini: Software for numerical algebraic geometry. Available at bertini.nd.edu.

[8] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. *Numerically solving polynomial systems with Bertini*, volume 25 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.

[9] A. M. Bruckstein, R. J. Holt, and A. N. Netravali. How to catch a crook. *J. Visual Communication and Image Representation*, 5(3):273–281, 1994.

[10] T. Chen, T.-L. Lee, and T.-Y. Li. Hom4PS-3: A parallel numerical solver for systems of polynomial equations based on polyhedral homotopy continuation methods. In H. Hong and C. Yap, editors, *Mathematical Software – ICMS 2014*, pages 183–190, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[11] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer, 1998.

[12] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2015.

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

[14] T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, and J. Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 2018.

[15] A. Ecker and A. D. Jepson. Polynomial shape from shading. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 145–152, June 2010.

[16] R. Fabbri. *Multiview Differential Geometry in Application to Computer Vision*. Ph.D. dissertation, Division Of Engineering, Brown University, Providence, RI, 02912, July 2010.

[17] R. Fabbri, P. J. Giblin, and B. B. Kimia. Camera pose estimation using first-order curve differential geometry. In *Proceedings of the IEEE European Conference in Computer Vision*, Lecture Notes in Computer Science. Springer, 2012.

[18] R. Fabbri and B. B. Kimia. Multiview differential geometry of curves. *International Journal of Computer Vision*, 117:1–23, 2016.

[19] O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge, MA, USA, 2001.

[20] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In G. Sandini, editor, *Computer Vision — ECCV'92*, pages 321–334, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[21] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1362–1376, 2010.

[22] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[23] J. D. Hauenstein and M. H. Regan. Adaptive strategies for solving parameterized systems using homotopy continuation. *Appl. Math. Comput.*, 332:19–34, 2018.

[24] A. Heyden. Reconstruction from image sequences by means of relative depths. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, pages 1058–, Washington, DC, USA, 1995. IEEE Computer Society.

[25] R. J. Holt and A. N. Netravali. Motion and structure from line correspondences: Some further results. *International Journal of Imaging Systems and Technology*, 5(1):52–61, 1994.

[26] R. J. Holt and A. N. Netravali. Number of solutions for motion and structure from multiple frame correspondence. *Int. J. Comput. Vision*, 23(1):5–15, May 1997.

[27] R. J. Holt, A. N. Netravali, and T. S. Huang. Experience in using homotopy methods to solve motion estimation problems. volume 1251, 1990.

[28] K. Huang, Y. Wang, Z. Zhou, T. Ding, S. Gao, and Y. Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018.

[29] B. Johansson, M. Oskarsson, and K. Astrom. Structure and motion estimation from complex features in three views. In *Proceedings of the Indian Conference on computer vision, graphics, and image processing*, 2002.

[30] L. F. Julià and P. Monasse. A critical review of the trifocal tensor estimation. In *Pacific-Rim Symposium on Image and Video Technology*, pages 337–349. Springer, 2017.

[31] L. Julià and P. Monasse. A critical review of the trifocal tensor estimation. In *The Eighth Pacific-Rim Symposium on Image and Video Technology – PSIVT'17*, Wuhan, China, 2017.

[32] Y. Kasten, M. Galun, and R. Basri. Resultant based incremental recovery of camera pose from pairwise matches. *CoRR*, abs/1901.09364, 2019.

[33] J. Kileel. Minimal problems for the calibrated trifocal variety. *SIAM Journal on Applied Algebra and Geometry*, 1(1):575–598, 2017.

[34] B. B. Kimia, X. Li, Y. Guo, and A. Tamrakar. Differential geometry in edge detection: accurate estimation of posi-

tion, orientation and curvature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[35] D. J. Kriegman and J. Ponce. Curves and surfaces. chapter A New Curve Tracing Algorithm and Some Applications, pages 267–270. Academic Press Professional, Inc., San Diego, CA, USA, 1991.

[36] D. J. Kriegman and J. Ponce. Geometric modeling for computer vision. volume 1610, 1992.

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

[38] Y. Kuang, M. Oskarsson, and K. Åström. Revisiting trifocal tensor estimation using lines. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2419–2423. IEEE, 2014.

[39] V. Larsson, K. Åström, and M. Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] V. Larsson, M. Oskarsson, K. Åström, A. Wallis, Z. Kukelova, and T. Pajdla. Beyond grobner bases: Basis selection for minimal solvers. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3945–3954, 2018.

[41] P. Lei, F. Li, and S. Todorovic. Joint spatio-temporal boundary detection and boundary flow prediction with a fully convolutional siamese network. CVPR, 2018.

[42] S. Leonardos, R. Tron, and K. Daniilidis. A metric parametrization for trifocal tensors with non-colinear pinholes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 259–267, June 2015.

[43] A. Leykin. Numerical algebraic geometry. *J. Softw. Alg. Geom.*, 3:5–10, 2011.

[44] 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.

[45] E. Martyushev. On some properties of calibrated trifocal tensors. *Journal of Mathematical Imaging and Vision*, 58(2):321–332, 2017.

[46] J. Mathews. Multi-focal tensors as invariant differential forms. *arXiv e-prints*, page arXiv:1610.04294, Oct 2016.

[47] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *Int. J. Comput. Vision*, 8(2):123–151, 1992.

[48] A. Morgan. *Solving polynomial systems using continuation for engineering and scientific problems*, volume 57 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009. Reprint of the 1987 original.

[49] P. K. Nanda, U. B. Desai, and P. Poonacha. A homotopy continuation method for parameter estimation in mrf models and image restoration. In *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, 1994.

[50] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[51] D. Nistér and F. Schaffalitzky. Four points in two or three calibrated views: Theory and practice. *Int. J. Comput. Vision*, 67(2):211–231, 2006.

[52] D. NistĂŠr, O. Naroditsky, and J. Bergen. Visual odometry. In *Computer Vision and Pattern Recognition (CVPR)*, pages 652–659, 2004.

[53] L. Oeding. The quadrifocal variety. *arXiv e-prints*, 2015.

[54] M. Oskarsson, A. Zisserman, and K. Astrom. Minimal projective reconstruction for combinations of points and lines in three views. *Image and Vision Computing*, 22(10):777 – 785, 2004. British Machine Vision Computing 2002.

[55] S. Petitjean. Algebraic geometry and computer vision: Polynomial systems, real and complex roots. *Journal of Mathematical Imaging and Vision*, 10(3):191–220, May 1999.

[56] S. Petitjean, J. Ponce, and D. J. Kriegman. Computing exact aspect graphs of curved objects: Algebraic surfaces. *International Journal of Computer Vision*, 9(3):231–255, Dec 1992.

[57] M. Pollefeys. Vnl realnpoly: A solver to compute all the roots of a system of n polynomials in n variables through continuation. Available at `https://github.com/vxl/vxl/blob/master/core/vnl/algo/vnl_rnpoly_solve.h`, 1997.

[58] M. Pollefeys and L. Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):707–724, Aug. 1999.

[59] A. Qadir and J. Neubert. A line-point unified solution to relative camera pose estimation. *CoRR*, abs/1710.06495, 2017.

[60] L. Quan, B. Triggs, and B. Mourrain. Some results on minimal euclidean reconstruction from four points. *J. Math. Imaging Vis.*, 24(3):341–348, 2006.

[61] L. Quan, B. Triggs, B. Mourrain, and A. Ameller. Uniqueness of minimal Euclidean reconstruction from 4 points. Technical report, 2003. unpublished article.

[62] 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.

[63] V. Rodehorst. Evaluation of the metric trifocal tensor for relative three-view orientation. In *International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, July 2015.

[64] M. Salzmann. Continuous inference in graphical models with polynomial energies. In *CVPR*, pages 1744–1751. IEEE Computer Society, 2013.

[65] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233, 2000.

[66] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[67] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision (IJCV)*, 80(2):189–210, 2008.

[68] A. J. Sommese and C. W. Wampler, II. *The numerical solution of systems of polynomials arising in engineering and science*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[69] C. Strecha, W. von Hansen, L. J. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[70] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[71] 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.

[72] A. Vakhitov, V. Lempitsky, and Y. Zheng. Stereo relative pose from line and point feature triplets. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[73] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, June 1999.

[74] J. Zhao, L. Kneip, Y. He, and J. Ma. Minimal case relative pose computation using ray-point-ray features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. DMS-1439786 while most authors were in residence at Brown University's Institute for Computational and Experimental Research in Mathematics – ICERM, in Providence, RI, during the Fall 2018 and Spring 2019 semesters.

Supplementary Material Appendix

A. Other formulations

Other "non-minor" formulations were also explored and implemented in MINUS, notably for Chicago. Two important formulations exist are worth mentioning. The first, is to eliminate depths and other scalar from the original equations from Section 2.1, ending with a 11×11 system of equations only in the relative poses $\mathtt{R}_1, \mathbf{t}_1, \mathtt{R}_2, \mathbf{t}_2$ *modulo* global scale – embodying the calibrated trifocal tensor in different forms depending on the representation employed. Another approach is further eliminating translations, to obtain 6×6 equation in $\mathtt{R}_2, \mathtt{R}_3$, which can give better performance in linear solves in Algorithm 1. These are explored in the present supplementary material along with more aggressive optimization strategies for an optimized solver with reliability tradeoffs.

Chicago Let $(\mathtt{R}_i, \mathbf{t}_i)$ for $i = 2, 3$ be the rotation matrix and translation vector for camera 1 relative to camera i. The parameters in local image coordinates are given as:

- point j in view i: $\mathbf{x}_{ij}$ for $i = 1, 2, 3, j = 1, 2, 3$ and
- tangent at point j in view i: $\mathbf{d}_{ij}$, $i = 1, 2, 3, j = 1, 2$.

The first set of vector equations is associated to viewing point $j = 1, 2, 3$ from cameras $i = 1, 2, 3$:

$$\alpha_{ij}\mathbf{x}_{ij} = \mathtt{R}_i \alpha_{1j}\mathbf{x}_{1j} + \mathbf{t}_i. \tag{9}$$

Eliminate $\mathbf{t}_i$ by using (9) for $j = 3$ and rearrange to:

$$\alpha_{ij}\mathbf{x}_{ij} - \alpha_{i3}\mathbf{x}_{i3} = \mathtt{R}_i(\alpha_{1j}\mathbf{x}_{1j} - \alpha_{13}\mathbf{x}_{13}), \quad i = 2, 3, \quad j = 1, 2. \tag{10}$$

The second set of vector equations is associated to viewing tangent j at point $j = 1, 2$ from cameras $i = 1, 2, 3$:

$$\epsilon_{ij}\mathbf{x}_{ij} + \mu_{ij}\mathbf{d}_{ij} = \mathtt{R}_i(\epsilon_{1j}\mathbf{x}_{1j} + \mu_{1j}\mathbf{d}_{1j}). \tag{11}$$

Together, (10) and (11) are a set of 24 scalar equations with the following unknowns:

$$(\mathtt{R}_i, \mathbf{t}_i), i = 2, 3; \alpha_{ij}, i = 1, 2, 3, j = 1, 2, 3;$$
$$(\epsilon_{ij}, \mu_{ij}), i = 1, 2, 3, j = 1, 2,$$

which are used in our additional Bertini solver in regards to the non-minor Chicago formulation and experimentation.

Cleveland The three labeled points are the same, therefore (9) and its reduction to (10) still applies. With the description in the main manuscript, for the free 3D line L, we let $(\mathbf{p}_v, \mathbf{q}_v)$ be two distinct points in three views. The back-projection of the image line is a plane whose equation in local coordinates is given in terms of a vector $\mathbf{n}_v$ normal to the plane:

$$\mathbf{n}_v^\top \mathbf{x} = 0, \text{ where } \mathbf{n}_v = \mathbf{p}_v \times \mathbf{q}_v.$$

Point $\mathbf{P}$ chosen as $\mathbf{P} = \alpha_p \mathbf{p}_1$, must lie in the back-projection planes in the other two views, giving

$$\mathbf{n}_v^\top (\mathtt{R}_v \alpha_p \mathbf{p}_1 + \mathbf{t}_v) = 0, \tag{12}$$

for views 2 and 3. Eliminating the translations and rearranging we have:

$$\alpha_{v3}\mathbf{n}_v^\top \mathbf{x}_{v3} = \mathbf{n}_v^\top \mathtt{R}_v(\alpha_{13}\mathbf{x}_{13} - \alpha_p \mathbf{p}_1). \tag{13}$$

In addition, L must lie in all three back-projection planes, thus

$$\mathbf{n}_1^\top \mathbf{v} = 0, \quad \mathbf{n}_n^\top \mathtt{R}_2 \mathbf{v} = 0, \quad \mathbf{n}_3^\top \mathtt{R}_3 \mathbf{v} = 0. \tag{14}$$

The solver referenced below for this non-minor Cleveland formulation is defined by the polynomial system of (10), (13), and (14) with the following unknowns:

$$\mathtt{R}_v, \;\; v = 2, 3; \alpha_{vj}, \;\; v = 1, 2, 3, j = 1, 2, 3; \alpha_p; \mathbf{v}.$$

Of course, we note that the above equations can partially be represented as determinants equal to zero; by non-minor we simply mean it is not focused on minors, but that they are a by-product of another type of geometrical reasoning.

B. Clarifying the proof of degrees

In the main paper, a proof regarding the number of 312 degrees and 216 for Chicago and Cleveland, respectively, was provided focusing on numerical arguments. These arguments are mathematically sound as there are strong guarantees for polynomial functions on the results of numerical tests when undertaken with certain provisions we described. In our main manuscript we also sketched how the proof would proceed by means of symbolic techniques. We now provide details on such a procedure, which is standard practice [12, 11].

To obtain the degree of the system, it is enough to give random values to all symbolic parameters (or coefficients), and then compute the degree of the resulting (specialised) system. This can be performed over $\mathbb{Q}$, as briefly described in the paper, or it may be more feasible to carry out computations modulo p, for a suitable prime number p. By making sure that the random values of the parameters are generic enough to be a representative of the general ones, and that the prime that we use is not a bad prime (for example that the modulo p operation does not kill terms of the polynomials), then the computation of the degree is as mathematically sound as an analytic-geometric proof by hand (which would be very hard for this problem size).

Once we compute, over $\mathbb{Q}$, a lexicographical Gröbner basis, its last polynomial is a univariate polynomial of degree D, which is the problem degree. For Chicago, $D = 312$ is obtained, and for cleveland $D = 216$. Let the single variable of this last univariate polynomial be x. By solving this polynomial by usual means, one backsubstitutes x and thus find a solution for the system. The procedure over the rationals is time consuming (several hours to days), so as a solver this generic symbolic method as such is not useful in practice beyond proofs and other analysis.

C. Additional Synthetic Experiments

Synthetic experiments were completed for the minor formulation of Cleveland discussed in Section 2 in the main manuscript, as well as the other formulations outlined above in A. These experiments are equivalent to those outlined in Section 4 in the main manuscript under the heading synthetic experiments.

For the three separate formulations, minor Cleveland and alternative Chicago and Cleveland, it was found that pose estimation errors are negligible as shown in Figures 11, 12, and 13, respectively.

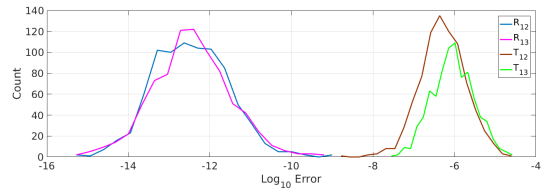

Figure 11. Errors of computed parameters w.r.t the GT are small showing that the solver is numerically stable of the minor formulation of Cleveland

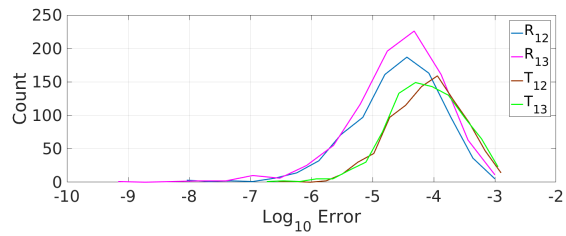

Figure 12. Errors of computed parameters w.r.t the GT are small showing that the solver is numerically stable of alternative formulation of Chicago

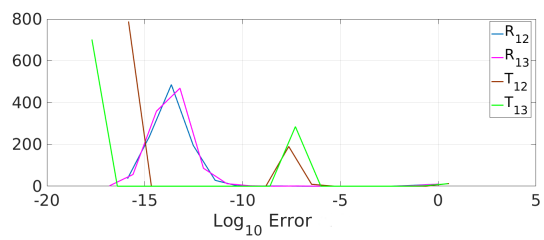

Figure 13. Errors of computed parameters w.r.t the GT are small showing that the solver is numerically stable of alternative formulation of Cleveland

The next set of experiments show the behavior when the correspondences are correct but noisy. Using the same process as described in detail in Experiment section in the main paper. The result of three different formulations (minor formulation of Cleveland, non-minor formulation of Chicago and non-minor formulation of Cleveland) are shown in Figure 14, 15, and 16. For each formulation, the median of the translation and rotation error are low. However, given the relatively high failure rate of these three formulations, we have several failure cases for each. But these failure cases can be detected by thresholding the maximum inlier ratio in RANSAC. The average reprojection error with respect to the ground truth point correspondences, also shown in Figure 14, 15, and 16, shows that for most of the test cases, we have a stable and reasonable reprojection error. Again, the case with large reprojection error can be ignored by thresholding maximum inlier ratio.

In addition, with the computational efficiency discussed below, these experiments with synthetic data and multiple formulations highlights the efficacy of the homotopy con-

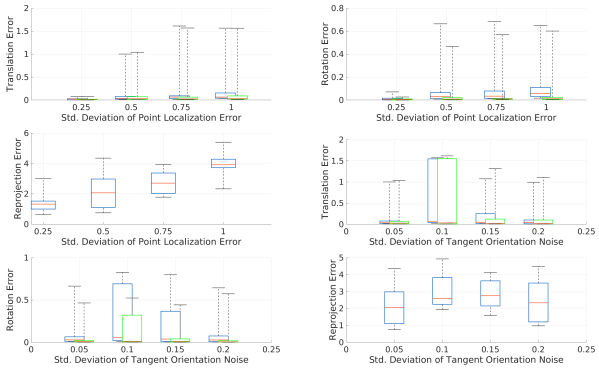

Figure 14. Distribution of trifocal pose error in the form of translational and rotational error plotted against the level of feature localization noise and orientation noise using the minor formulation of Cleveland. The green, resp., blue plots refer to the pose of the second resp., the third camera, relative to the first.

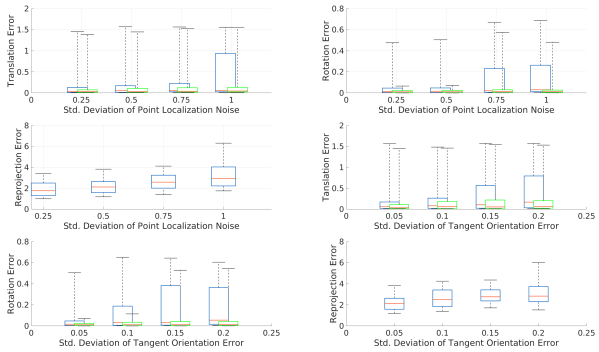

Figure 15. Distribution of trifocal pose error in the form of translational and rotational error is plotted against the level of feature localization noise and orientation noise with non-minor formulation of Chicago. The green, resp., blue plots refer to the pose of the second resp., the third camera, relative to the first.

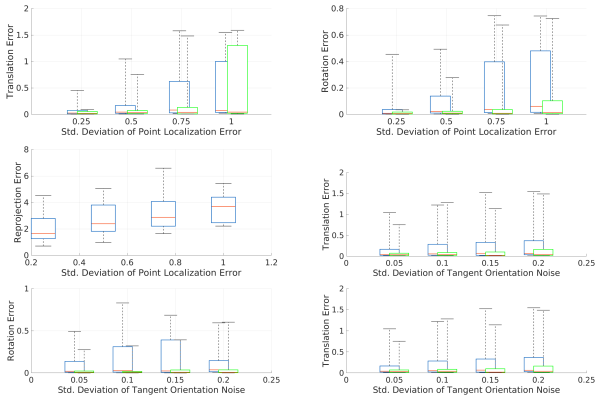

Figure 16. Distribution of trifocal pose error in the form of translational and rotational error is plotted against the level of feature localization noise and orientation noise with non-minor formulation of Cleveland. The green, resp., blue plots refer to the pose of the second resp., the third camera, relative to the first.

tinuation methods and their ability to solve these trifocal problems in a competitive nature.

Computational efficiency: For the minor formulation of Cleveland, each run of our more general purpose solver using Bertini takes about 8.97 seconds on average with a failure rate of about 17.9%. For the non-minor formulation of Cleveland, each run takes about 11.46 seconds on average with a failure rate of 3.2% and for the non-minor formulation of Chicago, each run takes about 19.69 seconds on average with a failure rate of 12.4%. All of these tests were done on an AMD Opteron 6378 2.4 GHz processor using 12 threads.

Implementation: The minor formulation of Cleveland and the non-minor formulations of both Chicago and Cleveland were implemented within a more general purpose solver involving Bertini. This software is used for the homotopy continuation solver in order to utilize the parameter homotopy method described in Algorithm 1 in the main paper. There are improvements that can be made to precision and error analysis using adaptive multiprecision path tracking [6], yet this comes at the expense of speed. In addition, other settings within Bertini can be employed, at the expense of reliability and causing a potential increase in failure rate. There is potential for other optimization, but that has not been explored here.

D. Tuning of the main solver MINUS

As stated in the main manuscript, MINUS can run at the milisecond scale with the 14×14 formulation, at the cost of increased failure rate. We have observed that in practice such failure rate might not be important for RANSAC, and can be controlled by performing tests to the input points and lines to rule out near-coplanar or near-collinear configurations which would make the system close to underconstrained.

In optimizing MINUS, one can constrain the number of iterations per solution path, which would yield the most effective speedup. Another important study is regarding the conditioning of the linearized homotopies (Jacobian matrices) as one varies the formulation. Yet another very promising idea is to vary the start system. Presently, the start system is precomputed from random parameters for the equations using monodromy. The start system can instead be sampled from the view-sphere for our synthetic data, and the closest camera could be selected matching a similar configuration of point-tangents.

In practice, we observed the following effective optimizations to the current code. First, the most important parameter to vary is the maximum number of correction steps (see Algorithm 1 in the paper); a maximum of 3 is the safe

default. Increasing it to anywhere from 4 to 7 gets the runtime down to 464ms. Another is the corrector tolerance: by increasing it 10000x, MINUS will run in 200ms. This parameter can be seen by inspecting our published source code. It affects how many correction iterations are performed. The error rate for these extreme cases of $200ms$ can be as high as 50%. However, we believe that by performing less strict tests focusing on reprojection error, this failure rate is significantly lower.

How to prune paths that take too long is definitely the next step for MINUS. Acceleration using SIMD has been studied, but by analyzing assembly output, most operations (complex vector multiplications and additions) are currently auto vectorized. Our tests point to the fact that reducing the representation to, say 6×6, as is ongoing at the present time, would provide strong improvements if ill-conditioning is taken care of. Our tests indicate that this would improve linear-algebra solves as well as evaluator lengths and instruction cache misses.

E. Additional Real Experiments

More real experiments that were not shown in main paper are shown in this section. For texture-rich images, more cases from EPFL dataset are first shown, followed with quantitative comparison with other trifocal methods, illustrating that our method is comparable or better than other trifocal methods. The results of more challenging scenes are also reported in this section.

More EPFL Results More sample results from EPFL datasets are shown in Figure 17. We compare with another trifocal method reported in [31], shown in Table 1. All the other methods are introduced in the reference. Since the referred paper just report on two sequences, Fountain P-11 and Herz-Jesu-P8; we also report this comparison with these two sequences.

Methods	R error (deg)	T error(deg)
TFT-L	0.292	0.638
TFT-R	0.257	0.534
TFT-N	0.337	0.548
TFT-FP	0.283	0.618
TFT-PH	0.269	0.537
MINUS (Ours)	0.137	0.673

Table 1. The pose error comparison between our method with other trifocal methods. Observe that our method has better rotation error and comparable translation error.

More Challenging Scenes As shown in Figure 1 in the main paper, we created a dataset of the scene of three mugs. This scene lacks point features which is hard for traditional structure from motion schemes. For these images, we put a calibration rig in the view for generating the ground truth and was removed manually when ground truth was generated. We built 20 triplets of images within this dataset. Within these 20 triplets, approximate camera poses of only 5 triplets can be generated with COLMAP; But with our method, 8 out of 20 approximate camera poses can be estimated. The rest of the images are lacking of reliable correspondence but with careful tuning of curve matching the success ratio is expected to become higher in the future. The sample successful cases are shown in Figure 18.

Figure 17. Trifocal relative pose estimation of EPFL dataset. For each row, image triplets samples are shown. The estimation results are shown on the right. Ground truth poses are in solid green and estimated poses are in red.

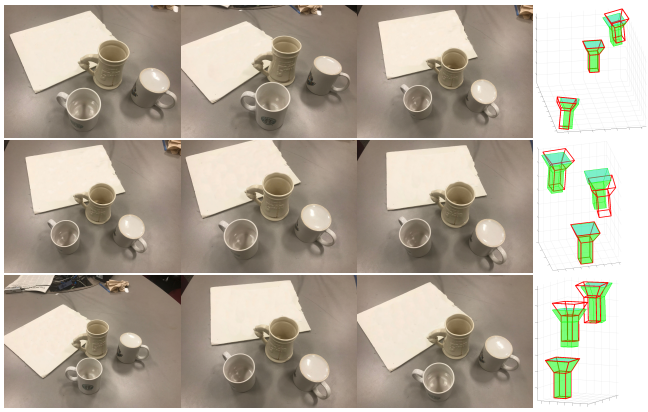

Figure 18. Trifocal relative pose estimation of EPFL dataset. For each row, image triplets samples are shown. The estimation results are shown on the right. Ground truth poses are in solid green and estimated poses are in red.