TRPLP – Trifocal Relative Pose from Lines at Points

Ricardo Fabbri[*]
Rio de Janeiro State University

Timothy Duff
Georgia Tech

Hongyi Fan
Brown University

Margaret H. Regan
University of Notre Dame

David da Costa de Pinho
UENF – Brazil

Elias Tsigaridas
INRIA Paris

Charles W. Wampler
University of Notre Dame

Jonathan D. Hauenstein
University of Notre Dame

Peter J. Giblin
University of Liverpool

Benjamin Kimia
Brown University

Anton Leykin
Georgia Tech

Tomas Pajdla
CIIRC CTU in Prague[†]

Abstract

We present a method for solving two minimal problems for relative camera pose estimation from three views, which are based on three view correspondences of (i) *three points and one line and* (ii) *three points and two lines through two of the points. These problems are too difficult to be efficiently solved by the state of the art Gröbner basis methods. Our method is based on a new efficient homotopy continuation (HC) solver, which dramatically speeds up previous HC solving by specializing HC methods to generic cases of our problems. We characterize their number of solutions and show with simulated experiments that our solvers are numerically robust and stable under image noise. We show in real experiments that* (i) *SIFT feature location and orientation provide good enough point-and-line correspondences for three-view reconstruction and* (ii) *that we can solve difficult cases with too few or too noisy tentative matches where the state of the art structure from motion initialization fails.*

1. Introduction

3D reconstruction has made an impact [4] by mostly relying on points in Structure from Motion (SfM) [1, 67, 23, 49]. Still, even production-quality SfM technology 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 moving cameras or objects; (*iv*) large scale changes where the overlap is not sufficiently significant; or (*v*) multiple and independently moving objects each lacking a sufficient number of features.

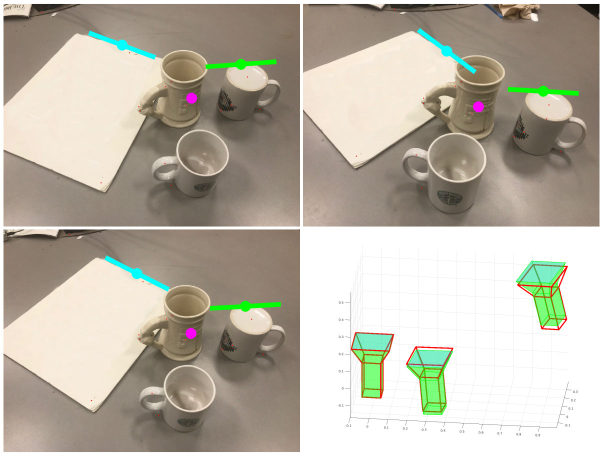

Figure 1. A deficiency of the traditional two-view approach to bootstraping SfM: not enough features detected (small red dots) and thus a SOTA SfM pipeline COLMAP [67] fails to reconstruct the relative camera pose. In contrast, the proposed trinocular method requires only three matching features: two triplets of point-tangents (points with SIFT orientation shown in green and cyan) and one triplet of points without orientation (purple) to reconstruct the pose. Red cameras are computed by our approach, and green shows ground truth.

The failure of bifocal pose estimation using RANSAC on hypothesized correspondences, *e.g.,* using 5 points [48], is highlighted in a dataset of images of mugs, Figure 1 (similar to the dataset in [51] but without a calibration board), for which the failure rate using the standard SfM pipeline COLMAP [63] is 75%. The failure of just directly apply-

[*]Contact: rfabbri@gmail.com, [†]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague. RF is supported by UERJ Prociência and FAPERJ Jovem Cientista do Nosso Estado E-26/201.557/2014. TD and AL are supported by NSF DMS-1151297. JDH and MHR are supported by NSF CCF-1812746, with additional support for JDH from ONR N00014-16-1-2722 and for MHR from Schmitt Leadership Fellowship in Science and Engineering. BK and HF are supported by the NSF grant IIS-1910530. TP is supported by the EU Regional Development Fund IMPACT CZ.02.1.01/0.0/0.0/15 003/0000468 and EU H2020 project ARtwin 856994. This work was initiated 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 (NSF DMS-1439786 and the Simons Foundation grant 507536).

ing the 5-point algorithm in this example is even higher. A similar situation exists for images containing repeated patterns where there are plenty of features, but determining correspondences is challenging. Most traditional multiview pipelines estimate the relative pose of the two best views and then register the remaining views using a P3P algorithm [68] to reduce the failure rate. The focus of this paper is to address the issue of failure of traditional bifocal algorithms in such cases.

The failure of bifocal algorithms motivates the use of (*i*) more complex features, *i.e.,* having additional attributes and (*ii*) more diverse features. We propose that *orientation* (in the sense of inclination) is a key attribute to disambiguate correspondences and we show that SIFT orientation in particular is a stable feature across views for trifocal pose estimation. Orientation can also from curve tangents [18, 17, 6], and the *orientation* of a straight line in multiple views also constrains pose. Observe, however, that orientation cannot be constrained in two views alone: SIFT orientation or line orientations in two views are uncorrelated, but together can identify their 3D counterparts and thus can constrain orientation in a third view. This motivates *trinocular pose estimation* based on point features endowed with orientation or including straight line features.

Camera estimation from trifocal tensors is long believed to augment two-view pose estimation [21], although a recent study suggests no significant improvements over bifocal pairwise estimation [31]. The calibrated trinocular relative pose estimation from four points, 3v4p, is notably difficult to solve [50, 59, 60, 17], and is not a minimal problem – it is over-constrained. The first working trifocal solver [50] effectively parametrizes the relative pose between two cameras as a curve of degree ten representing possible epipoles. A third view is then used to select the epipole that minimizes reprojection errors. In this sense, trinocular pose estimation has not truly been tackled as a minimal problem.

Trifocal pose estimation requires the determination of 11 degrees of freedom: six unknowns for each pair of rotation $\mathtt{R}$ and translation $\mathbf{t}$, less one for metric ambiguity. Three types of constraints arise in matching triplets of point features endowed with orientation. First, the epipolar constraint provides an equation for each pair of correspondences in two views. Second, in a triplet of correspondences, each pair of correspondences are required to match scale, providing another constraint; a total of three equations per triplet. It is easy to see, informally, that three points are insufficient to determine trifocal pose, while four points are too many. Third, each triplet of oriented feature points provides one orientation constraint. Thus, with three points, only two points need to be endowed with orientation, giving a total of 11 constraints for the 11 unknowns. We refer to this problem of three triplets of corresponding points, with two of the points having oriented features as "*Chicago*." In the second scenario, *i.e.*, using straight lines as features, with three points, only one free (unattached to a point) straight line feature is required. We refer to the problem of three triplets of corresponding points and one triplet of corresponding free lines as "*Cleveland*." This paper addresses trifocal pose estimation for the above two scenarios, shows that both are minimal problems, and develops efficient solvers for the resulting polynomial systems.

Specifically, each problem comprises eleven trifocal constraints that in principle give systems of eleven polynomials in eleven unknowns. These systems are not trivial to solve and require techniques from numerical algebraic geometry [9, 14, 41] *(i)* to probe whether the system is over or under constrained or otherwise minimal; *(ii)* to understand the range of the number of real solutions and estimate a *tight* upper bound; and *(iii)* to develop efficient and practically relevant methods for finding solutions which are real and represent camera configurations. This paper shows that the Chicago problem is minimal and has up to 312 solutions (the area code of Chicago is 312) of which typically 3-4 end up becoming relevant to camera configurations. Similarly, we show that the Cleveland problem is minimal and has up to 216 solutions. The minimality of combinations of points and lines for the general case [15] is a parallel development to the more concrete treatment presented here.

The numerical solution of polynomial systems with several hundred solutions is challenging. We devised a custom-optimized *Homotopy Continuation (HC)* procedure which iteratively tracks solutions with a guarantee of global convergence [14]. Our framework specializes the general HC approach to minimal problems typical of multiple view geometry, thereby dramatically speeding up the implementation. Specifically, our Chicago and Cleveland solvers are not only the first solvers for such high degree problems, but are orders of magnitude faster than solvers for such scale of problems: 660ms on average on an Intel core i7-7920HQ processor with four threads. They share the same generic core procedure with plenty of room to be further optimized for specific applications. Most significantly, since finding each solution is a completely independent integration path from the others, the solvers are suitable for implementation on a GPU, as a batch for RANSAC, which would then reduce the run time by the number of tracks, *i.e.,* by two orders of magnitude. We hope that our developments can be a template for solving other computer vision problems involving systems of polynomials with a large number of solutions, and in fact the provided C++ framework is fully templated to include new minimal problems seamlessly.

It should be emphasized that trifocal pose estimation as a more expensive operation is not intended as a competitor of bifocal estimation algorithms. Rather, the trifocal approach can be considered as a fallback option in situations where bifocal pose estimation fails.

Experiments are initially reported on complex synthetic data to demonstrate that the system is robust and stable under spatial and orientation noise and under a significant level of outliers. Experiments on real data first demonstrates that SIFT orientation is a remarkably stable cue over a wide variation in view. We then show that our approach is successful in all cases where the traditional SfM pipeline succeeds, but of course at higher computational cost. What is critically important is that the proposed approach succeeds in many other cases where the SfM pipeline fails, *e.g.,* on the EPFL [70] and Amsterdam Teahouse datasets [71], as shown in Figures 9 and 10. Those cases where the bifocal scheme fails – flagged by the number of inliers, for example – can consider the application of a currently more expensive but more capable trifocal scheme to allow for reconstructions that would otherwise be unsolved.

1.1. Literature Review

Trifocal Geometry Calibrated trifocal geometry estimation is a hard problem [50, 59, 60, 62]. There are no publicly available solvers we are aware of. The state of the art solver [50], based on four corresponding points (3v4p), has not yet found many practical applications [37].

For the uncalibrated case, 6 points are needed [26], and Larsson *et al.* recently solved the longstanding trifocal minimal problem using 9 lines [38]. The case of mixed points and lines is less common [53], but has seen a growing interest in related problems [63, 58, 72]. The calibrated cases beyond 3v4p are largely unsolved, spurring more sophisticated theoretical work [2, 3, 33, 40, 43, 44, 52]. 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, *i.e.,*, a given set of polynomial equations provably necessary and sufficient to describe calibrated trifocal geometry, is currently unknown.

Seminal works used curves and edges in three views to transfer differential geometry for matching [5, 61], and for pose and trifocal tensor estimation [13, 66], beyond straight lines for uncalibrated [24, 7] and calibrated [64, 63] SfM. Point-tangents – not to be confused with point-rays [11] – can be framed as *quivers* (1-quivers), or feature points with attributed directions (*e.g.,* corners), initially proposed in the context of uncalibrated trifocal geometry but de-emphasizing the connection to tangents to general curves [30, 74]. We note that point-tangent fields may also be framed as vector fields, so related technology may apply to surface-induced correspondence data [17]. In the calibrated setting, point-tangents were first used for absolute pose estimation by Fabbri *et al.* [18, 19], using only two points, later relaxed for unknown focal length [36]. The trifocal problem with three point-tangents as a local version of trifocal pose for global curves was first formulated by Fabbri [17], presented here as a minimal version codenamed Chicago.

Homotopy Continuation The basic theory of polynomial homotopy continuation (HC) [9, 46, 69] was developed in 1976, and guarantees algorithms that are *globally* convergent with probability one from given start solutions. A number of general-purpose HC softwares have considerably evolved over the past decade [8, 12, 41, 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, and pose estimation [10, 22, 27, 28, 29, 34, 35, 42, 45, 54, 55, 57], as well as for MRFs [10, 47], and in more recent work [16, 25, 65]. An implementation of the early continuation solver of Kriegman and Ponce [34] by Pollefeys is still widely available for low degree systems [56].

As an early example [27], 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 compared 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 [69]).

2. Two Trifocal Minimal Problems

2.1. Basic Equations

Our notation follows [24] with explicit projective scales. A more elaborate notation [13, 18] can be used to express the equations in terms of tangents to curves.

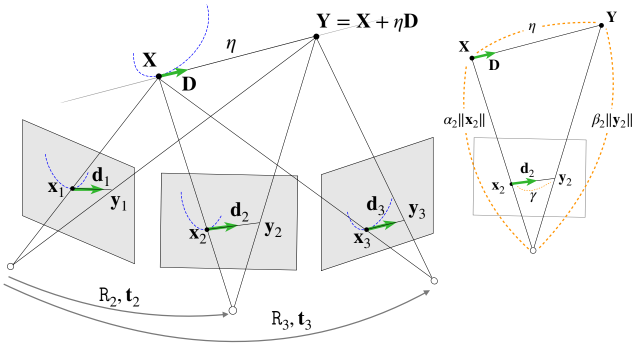

Figure 2. Notation for the trifocal pose problems.

Let $\mathbf{X}$ and $\mathbf{Y}$ denote inhomogeneous coordinates of 3D points and $\mathbf{x}_{pv}, \mathbf{y}_{pv} \in \mathbb{P}^2$ denote homogeneous coordinates of image points. Subscript p numbers the points and v numbers the views. If only a single subscript is used, it indexes views. Symbols $\mathtt{R}_v, \mathbf{t}_v$ denote the rotation and translation transforming coordinates from camera 1 to camera v; $\mathbf{d}$ 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 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 the new minimal problem, 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_{2v} in views $v = 1, 2, 3$, such that ℓ_{pv} meet $\mathbf{x}_{pv}$ for $p = 1, 2$, $v = 1, 2, 3$, 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

$$\alpha_v \mathbf{x}_v = \mathtt{R}_v \alpha_1 \mathbf{x}_1 + \mathbf{t}_v, \quad v = 2, 3. \tag{1}$$

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

$$\beta_v \gamma_v \mathbf{d}_v = \mathbf{y}_v - \mathbf{x}_v = \alpha_v \mathbf{x}_v + \eta \mathbf{D} - \mathbf{x}_v, \tag{3}$$

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 $v = 2, 3$. To simplify notation further, we change variables as $\epsilon_v = \beta_v - \alpha_v$, $\mu_v = \beta_v \gamma_v$ and get

$$\epsilon_v \mathbf{x}_v + \mu_v \mathbf{d}_v = \mathtt{R}_v \left(\epsilon_1 \mathbf{x}_1 + \mu_1 \mathbf{d}_1 \right), \quad v = 2, 3. \tag{5}$$

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

$$\operatorname{rank} \left[[I \mid 0]^T \ell_1 \mid [\mathtt{R}_2 \mid \mathbf{t}_2]^T \ell_2 \mid [\mathtt{R}_3 \mid \mathbf{t}_3]^T \ell_3 \right] < 3. \tag{6}$$

Equations 1 and 6 are the basic equations for Cleveland.

There are many ways to use elimination from these basic equations to obtain alternate formulations for these problems. A particular formulation based on vanishing minors for both Chicago and Cleveland, which produced our first working solver for 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 aim to determine the relative poses of the cameras. The concatenated list of all the feature 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 in projective space. For N cameras, the combined poses of cameras $2, \ldots, N$ relative to camera 1 are points 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 by considering 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.

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

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 the condition. 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's enough to find $(w, x) \in W \times X$ where the Jacobian of $V(w, x)$ is full rank. Such a rank test for a random point (w, x) serves to establish nonsingularity with probability one. Using floating point arithmetic this is highly indicative but not rigorous unless one bounds floating-point error, which can be done using interval or exact arithmetic. A singular value decomposition of the Jacobian using floating point showing 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.

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, but the definite proof by computer involves symbolically computing the Gröbner basis over $\mathbb{Q}$, with special provisions, as discussed in the 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.

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 equations based on minors described in 3.1, while other formulations are discussed in supplementary 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 [8, 41], 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 for the experiments described in section 4.

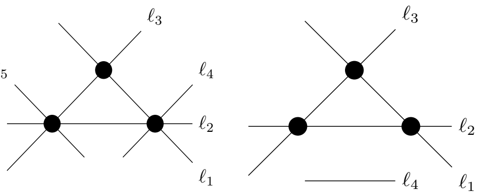

Figure 3. Visible line diagrams for Chicago and Cleveland.

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 $\ell_{1v}, \ldots, \ell_{5v} \in \mathbb{C}^{3\times 1}$ for each $v \in \{1, 2, 3\}$. 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

$$L_j = \left[[I\,|0]^T \ell_{j1} \quad [\mathtt{R}_2\,|\mathbf{t}_2]^T \ell_{j2} \quad [\mathtt{R}_3\,|\mathbf{t}_3]^T \ell_{j3} \right], \qquad (7)$$

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

We may describe the Cleveland problem with similar equations. For this problem, we are given lines $\ell_{1v}, \ldots, \ell_{4v}$ for $v \in \{1, 2, 3\}$. We enforce line correspondences for matrices $L_1, \ldots, L_4$ defined as in (7) 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.

3.2. Algorithm

From the previous section, we may define a specific system of polynomials $F(\mathcal{R}; \mathcal{A})$ in the unknowns $\mathcal{R} = (\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3)$ parametrized by $\mathcal{A} = (\ell_{11}, \ldots)$. Many representations for rotations were explored, but our main implementation employs quaternions. A fundamental technique for solving such systems, fully described in [69], 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 $\mathcal{A}_0$ 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 $F(\mathcal{R}; \mathcal{A}_0) = 0$, where $\mathcal{A}_0$ are randomly generated start parameters [9, 69]. 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.

Algorithm 1: Homotopy continuation solution tracker

```
input: Polynomial system $F(\mathcal{R}; \mathcal{A})$, where
        $\mathcal{R} = (\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3)$, and $\mathcal{A}$ parametrizes the data;
        Start parameters $\mathcal{A}_0$; start solutions $\mathcal{R}_0$ where
        $F(\mathcal{R}_0; \mathcal{A}_0) = 0$; Target parameters $\mathcal{A}^*$
output: Set of target solutions $\mathcal{R}^*$ where $F(\mathcal{R}^*; \mathcal{A}^*) = 0$

Setup homotopy $H(\mathcal{R}; s) = F(\mathcal{R}; (1-s)\mathcal{A}_0 + s\mathcal{A}^*)$.
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 $\mathcal{R}^*$ where $H(\mathcal{R}^*, 1) = 0$.
```

3.3. Implementation

We provide an optimized open source C++ package called MINUS – MInimal problem NUmerical Solver[1]. This is a homotopy continuation code specialized for minimal problems, templated in C++, so that efficient specialization for different problems and different formulations are 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 of magnitude, there may be reliability tradeoffs (*c.f.* supplementary material).

4. Experiments

Experiments are conducted first for synthetic data for a detailed and controlled study, followed by experiment on challenging real data. Due to space constraints, we present results for the more challenging Chicago problem, leaving Cleveland for supplementary materials.

Synthetic data experiments: The synthetic data from [20, 18] consists of 3D curves in a $4 \times 4 \times 4cm^3$ volume projected to 100 cameras (Figure 4), and sampled to get 5117 points enclosed with orientations (tangents of curves) that are projections of the same 3D analytic points and 3D curve tangents [20], and then degraded with noise and outliers. Camera centers are randomly sampled around an average sphere around the scene along normally distributed radii of mean 1m and $\sigma = 10mm$. Rotations are 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. 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.

The error in pose estimation is compared with ground-truth as the angular error between normalized translation vectors and the angular error between the quaternions. The process of generating the input to pose computation is repeated 1000 times and averaged. This experiment demonstrates that: (*i*) pose estimation errors are negligible, Figure 5(a); (*ii*) the number of real solutions is small: 35 real solutions on average, pruned down to 7 on average by enforcing positive depth, and even further to about 3-4 physically realizable solutions on average employing the unused tangent of the third point as verification, Figure 5(b); (*iii*) the solver fails in about 1% of cases, which are detectable and, while not a problem for RANSAC, can be eliminated by running the solver 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 camera 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 $\sigma \in \{0.25, 0.5, 0.75, 1.0\}$ pixels and to the orientation of the tangents with $\sigma \in \{0.05, 0.1, 0.15, 0.2\}$ radians, reflecting expected feature localization and orientation localization error. A RANSAC scheme determines the feature set that generates the highest number of inliers. Experiments indicate that the translation and rotation errors are reasonable. Figure 6 (top) shows how the extent of localization error affects pose (in terms

[1] Code available at `http://github.com/rfabbri/minus`

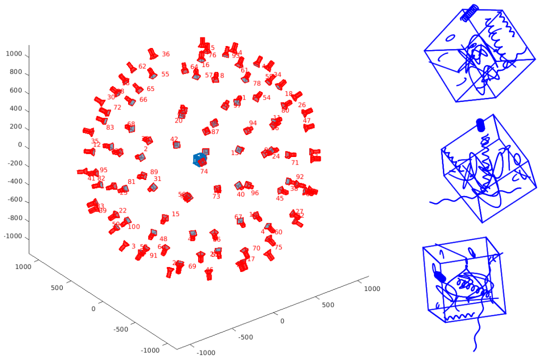

Figure 4. Sample views of our synthetic dataset. Real datasets have also been used in our experiments. (3D curves are from [18, 20]).

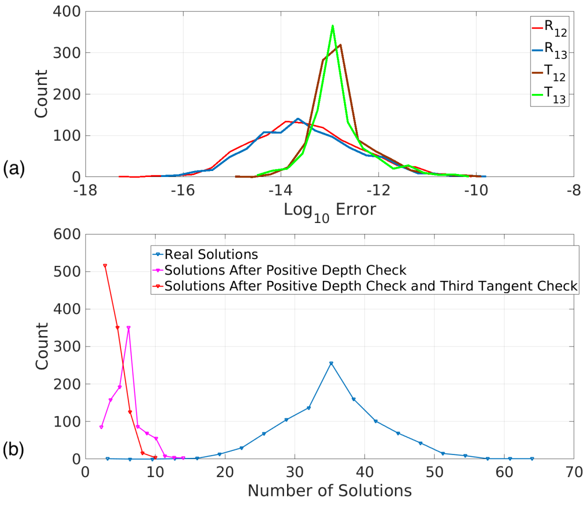

Figure 5. (a) Errors of computed pose are small showing that the solver is numerically stable. (b) The histogram of the numbers of real solutions in different stages.

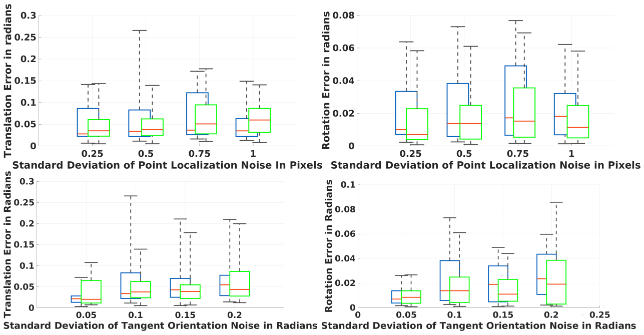

Figure 6. Translational and rotational error distributions between cameras 1 and 2 (blue) and 1 and 3 (green) for different levels of feature localization (top) and orientation noise (bottom).

of translation and rotation errors) under a fixed orientation perturbation of 0.1 radians; Figure 6 (bottom) shows how the extent of orientation error affects pose under a fixed localization error of 0.5 pixels. The more meaningful reprojection error, *i.e.*, the distance of a point from the location

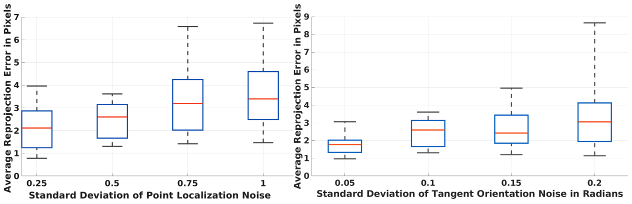

Figure 7. Distributions of reprojection error of feature location plotted against localization and orientation errors.

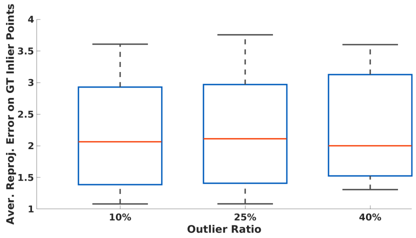

Figure 8. Average reprojection error on ground truth inlier points with different ratio of outliers.

determined by the other two points in a triplet, is shown in Figure 7, averaged over 100 triplets.

The third experiment probes whether the system can reliably and accurately determine trifocal pose when correct noisy correspondences are mixed with outliers. With a fixed feature localization error of 0.25 pixels and feature orientation error of 0.1 radians, 200 triplets of features were generated, with a percentage of these replaced with samples having random location and orientation. The ratio of outliers is varied over 10%, 25% and 40%, and the experiment is repeated 100 times for each. The resulting reprojection error is small and stable across outlier ratios, Figure 8.

Computational efficiency: Each solve using our software MINUS takes $660ms$ on average (1.9s in the worst case) as compared to over 1 minute on average for the best prototypes using general purpose software [8, 41], 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 data experiments: 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 the relative pose of the three cameras is determined from two tangents with their assigned SIFT orientation and a third point without orientation.

Figure 9 shows that camera pose is reliably and accurately found using triplets of images taken from the EPFL dense multi-view stereo test image dataset [70]. Our quantitative estimates on 150 random triplets from this dataset give pose errors of 1.5×10^{-3} radians in translation and 3.24×10^{-4} radians in rotation. The average reprojection error is 0.31 pixels. These are comparable to or better than the trifocal relative pose estimation methods reported in [31]. Our conclusion for this dataset, whose purpose is simply to validate the solver, is that our method is at least as good and often better than the traditional ones. See supplementary data for more examples and a substantiation of this claim. Note that we do not advocate replacing the traditional method for this dataset. We simply state that our method works just as well, of course at a higher cost.

The EPFL dataset is feature-rich, typically yielding on the order of 1000 triplet features per image triplet. As such it does not portray some of the typical problems faced in challenging situations when there are few features available. The Amsterdam Teahouse Dataset [71], which also has ground-truth relative pose data, depicts scenes with fewer features. Figure 10 (top) shows a triplet of images from this dataset where there is a 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 [67]). However, when the number of features is reduced, as in Figure 10 (bottom) where the soup can is occluded, COLMAP fails to find the relative pose between pairs of these images. In contrast, our approach, which relies on three and not five features, is able to recover the camera pose for this scene. Further results are in supplementary material.

We also created another featureless dataset similar to the one in [51] but with the calibration board manually removed. This scene lacks point features, which is extremely challenging for traditional structure from motion. We built 20 triplets of images within this dataset. Within these 20 triplets, camera poses of only 5 triplets can be generated with COLMAP, but with our method, 10 out of 20 camera poses can be estimated. We reached a 100% improvement over the standard pipeline on image triplets. The sample successful cases are shown in Figure 1 and 11.

5. Conclusion

We presented a new calibrated trifocal minimal problem, an analysis demonstrating its number of solutions, and a practical solver by specializing general computation techniques from numerical algebraic geometry. Our approach is able to characterize and solve a similar difficult minimal problem with mixed points and lines. The increased ability to solve trifocal problems is key to future work on broader problems connecting the multi-view geometry of points and lines to that of points and tangents appearing when observing 3D curves, *e.g.*, in scenes without point features, using tools of differential geometry [17, 20]. Our "100 lines of custom-made solution tracking code" will also be used to try to improve solvers of many other minimal problems which have not been solved efficiently with Gröbner bases [39].

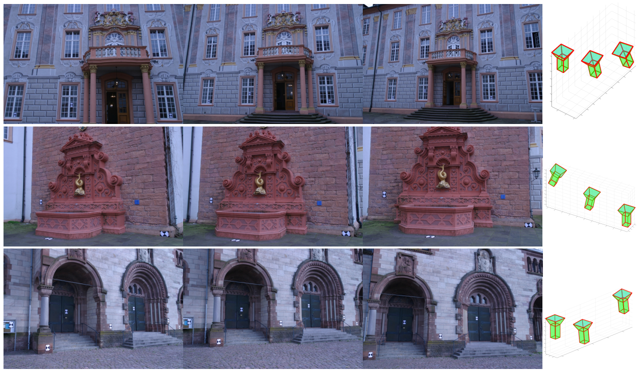

Figure 9. Trifocal relative pose estimation of EPFL dataset. At each row, image samples are shown with results on the right: ground truth in green and estimated poses in red outlines.

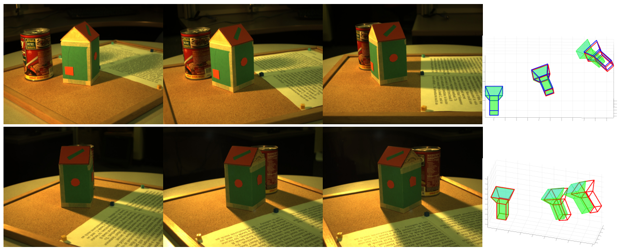

Figure 10. Samples of trifocal relative pose estimation of the Amsterdam Teahouse dataset. Top row is a sample triplet of images that COLMAP is able to tackle; second row is a triplet from the images where COLMAP fails. COLMAP results are in blue outlines, our results are in red, and ground truth is green.

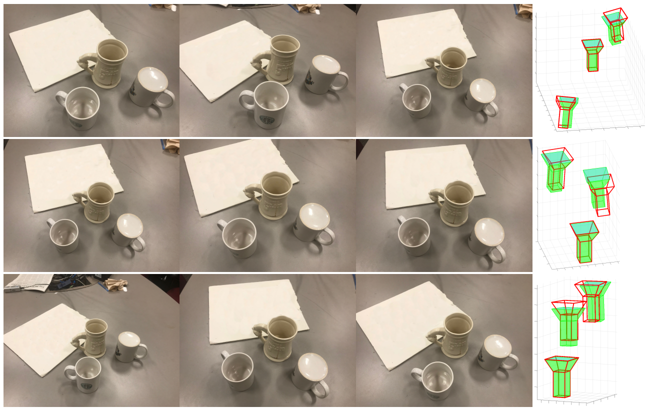

Figure 11. Trifocal relative pose results for a dataset comprising three mugs, which is challenging for traditional SfM. For each row, image triplet samples are shown, with results on the right. Ground truth poses are in solid green and estimated poses are in red.

References

[1] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 2009.

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

[3] Alberto Alzati and Alfonso 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 1^{st} *International Conference on Computer Vision*, June 1987.

[6] Daniel Barath and Zuzana Kukelova. Homography from two orientation- and scale-covariant features. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[7] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer vision and image understanding*, 100(3):416–441, 2005.

[8] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Bertini: Software for numerical algebraic geometry. Available at `bertini.nd.edu`.

[9] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles 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.

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

[11] Federico Camposeco, Torsten Sattler, and Marc Pollefeys. Minimal solvers for generalized pose and scale estimation from two rays and one point. In *European Conference on Computer Vision*, pages 202–218. Springer, 2016.

[12] Tianran Chen, Tsung-Lin Lee, and Tien-Yien Li. Hom4PS-3: A parallel numerical solver for systems of polynomial equations based on polyhedral homotopy continuation methods. In Hoon Hong and Chee Yap, editors, *Mathematical Software – ICMS 2014*, pages 183–190, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

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

[14] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 39(3):1421–1446, 2018.

[15] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomas Pajdla. Plmp-point-line minimal problems in complete multi-view visibility. *arXiv preprint arXiv:1903.10008*, 2019.

[16] 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.

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

[18] Ricardo Fabbri, Peter J. Giblin, and Benjamin 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.

[19] Ricardo Fabbri, Peter J. Giblin, and Benjamin B. Kimia. Camera pose estimation using first-order curve differential geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. Accepted.

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

[21] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. MIT Press, Cambridge, MA, USA, 2001.

[22] 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.

[23] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, Aug. 2010.

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

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

[26] 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.

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

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

[29] Robert J. Holt, Arun N. Netravali, and Thomas S. Huang. Experience in using homotopy methods to solve motion estimation problems. volume 1251, 1990.

[30] 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.

[31] Laura Julià and Pascal Monasse. A critical review of the trifocal tensor estimation. In *The Eighth Pacific-Rim Symposium on Image and Video Technology – PSIVT'17*, pages 337–349, Wuhan, China, 2017. Springer.

[32] Yoni Kasten, Meirav Galun, and Ronen 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] David J. Kriegman and Jean 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.

[35] David J. Kriegman and Jean Ponce. Geometric modeling for computer vision. volume 1610, 1992.

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

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

[38] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomás 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.

[40] 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.

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

[42] 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.

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

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

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

[46] Alexander 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.

[47] Pragyan K. Nanda, Uday B. Desai, and P.G. 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.

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

[49] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition (CVPR)*, pages 652–659, 2004.

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

[51] Irina Nurutdinova and Andrew 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*, pages 2363–2371, 2015.

[52] Luke Oeding. The quadrifocal variety. *arXiv e-prints*, 2015.

[53] Magnus Oskarsson, Andrew Zisserman, and Kalle 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.

[54] 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.

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

[56] Marc Pollefeys. VNL RealNPoly: A solver to compute all the roots of a system of n polynomials in n variables through continuation. Available at `github.com/vxl/vxl/blob/master/core/vnl/algo/` source code file `vnl_rnpoly_solve.h`, 1997.

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

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

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

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

[61] 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.

[62] 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.

[63] Yohann Salaün, Renaud Marlet, and Pascal Monasse. Robust and accurate line-and/or point-based pose estimation without manhattan assumptions. In *European Conference on Computer Vision*, pages 801–818. Springer, 2016.

[64] Yohann Salaün, Renaud Marlet, and Pascal Monasse. Line-based robust SfM with little image overlap. In *2017 International Conference on 3D Vision (3DV)*, pages 195–204. IEEE, 2017.

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

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

[67] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

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

[70] Christoph Strecha, Wolfgang von Hansen, Luc J. Van Gool, Pascal Fua, and Ulrich 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.

[71] Anil Usumezbas, Ricardo Fabbri, and Benjamin B. Kimia. From multiview image curves to 3D drawings. In *Proceedings of the European Conference in Computer Visiohn*, 2016.

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

[73] Jan 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.

[75] Ji Zhao, Laurent Kneip, Yijia He, and Jiayi Ma. Minimal case relative pose computation using ray-point-ray features. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

TRPLP – Trifocal Relative Pose from Lines at Points

Supplementary Material

Ricardo Fabbri[*]
Rio de Janeiro State University

Timothy Duff
Georgia Tech

Hongyi Fan
Brown University

Margaret H. Regan
University of Notre Dame

David da Costa de Pinho
UENF – Brazil

Elias Tsigaridas
INRIA Paris

Charles W. Wampler
University of Notre Dame

Jonathan D. Hauenstein
University of Notre Dame

Peter J. Giblin
University of Liverpool

Benjamin Kimia
Brown University

Anton Leykin
Georgia Tech

Tomas Pajdla
CIIRC CTU in Prague[†]

A. Other formulations

Along with the two minor based formulations described in the main manuscript, two alternate formulations of both Chicago and Cleveland were explored, as outlined below. Experimental results using synthetic data for these two alternate formulations of Chicago and Cleveland, as well as the minor formulation of Cleveland are discussed below in Section C.

In addition, other "non-minor" formulations of Chicago were explored and implemented in MINUS for optimization and testing. Two important formulations are worth mentioning. The first is obtained by eliminating depths and other scalars from the original equations from Section 2.1 of the main paper, ending with an 11×11 system of equations only in the relative poses $\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3$ *modulo* global scale – embodying the calibrated trifocal tensor in different forms depending on the representation employed. The second reduction occurs after further eliminating translations to obtain a 6×6 system of equations in $\mathtt{R}_2, \mathtt{R}_3$, which can give better performance for the linear solves within Algorithm 1. The results of using these formulations and other more aggressive optimization strategies within MINUS are outlined below in Section D.

Alternate Chicago The first set of vector equations, (1) in the main paper, is associated to viewing points $p = 1, 2, 3$ from cameras $v = 1, 2, 3$. Eliminate $\mathbf{t}_v$ using one such equation for $p = 3$ and rearrange to:

$$\alpha_{pv}\mathbf{x}_{pv} - \alpha_{3v}\mathbf{x}_{3v} = \mathtt{R}_v(\alpha_{p1}\mathbf{x}_{p1} - \alpha_{31}\mathbf{x}_{31}), \qquad (1)$$

for $v = 2, 3$ and $p = 1, 2, 3$. The second set of vector equations used by this formulation is associated to viewing tangents from cameras $v = 1, 2, 3$, which is (5) in the main paper. Together, (1) above and (5) of the main paper are a set of 24 scalar equations with the following unknowns:

$$(\mathtt{R}_v, \mathbf{t}_v),\, v = 2, 3; \quad \alpha_{pv},\, v = 1, 2, 3,\ p = 1, 2, 3;$$
$$(\epsilon_{pv}, \mu_{pv}),\, v = 1, 2, 3,\ p = 1, 2,$$

which are used with our additional Bertini solver in regards to the non-minor (*i.e.*, without using determinantal "visible lines" formulation of Section 3.1 of the main paper) Chicago formulation and experimentation.

Alternate Cleveland The three labeled points are the same, therefore (1) 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 $\mathbf{P} = \alpha_q \mathbf{p}_1$, must lie in the back-projection planes in the other two views, giving

$$\mathbf{n}_v^\top(\mathtt{R}_v \alpha_q \mathbf{p}_1 + \mathbf{t}_v) = 0, \qquad (2)$$

[*] Contact: rfabbri@gmail.com, [†] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague. RF is supported by UERJ Prociência and FAPERJ Jovem Cientista do Nosso Estado E-26/201.557/2014. TD and AL are supported by NSF DMS-1151297. JDH and MHR are supported by NSF CCF-1812746, with additional support for JDH from ONR N00014-16-1-2722 and for MHR from Schmitt Leadership Fellowship in Science and Engineering. BK and HF are supported by the NSF grant IIS-1910530. TP is supported by the EU Regional Development Fund IMPACT CZ.02.1.01/0.0/0.0/15 003/0000468 and EU H2020 project ARtwin 856994. This work was initiated 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 (NSF DMS-1439786 and the Simons Foundation grant 507536).

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

$$\alpha_{3v}\mathbf{n}_v^\top \mathbf{x}_{3v} = \mathbf{n}_v^\top \mathtt{R}_v(\alpha_{31}\mathbf{x}_{31} - \alpha_q \mathbf{p}_1). \tag{3}$$

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

$$\mathbf{n}_1^\top \mathbf{v} = 0, \quad \mathbf{n}_2^\top \mathtt{R}_2 \mathbf{v} = 0, \quad \mathbf{n}_3^\top \mathtt{R}_3 \mathbf{v} = 0. \tag{4}$$

The additional solver referenced below for this non-minor Cleveland formulation is defined by the polynomial system of (1), (3), and (4) with the following unknowns:

$$\mathtt{R}_v,\ \ v = 2,3;\ \ \alpha_q;\ \ \mathbf{v};$$
$$\alpha_{pv},\ \ p = 1,2,3,\ \ v = 1,2,3.$$

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 due to guarantees on the behavior of polynomial systems for these numerical methods given our assumptions listed within Section 2.2 of the main manuscript. 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 [2, 3].

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 (specialized) 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), 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 finds 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.

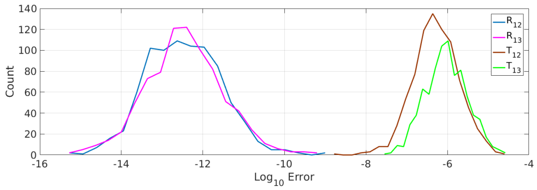

Figure 1. Errors of computed parameters with respect to the ground truth are small showing that the solver is numerically stable for the minor formulation of Cleveland.

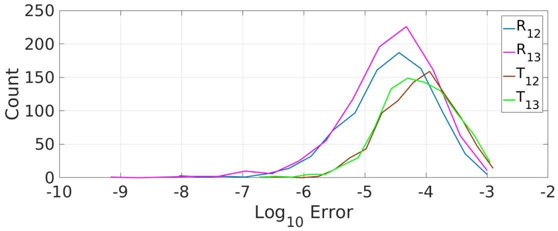

Figure 2. Errors of computed parameters with respect to the ground truth are small showing that the solver is numerically stable for the alternate formulation of Chicago.

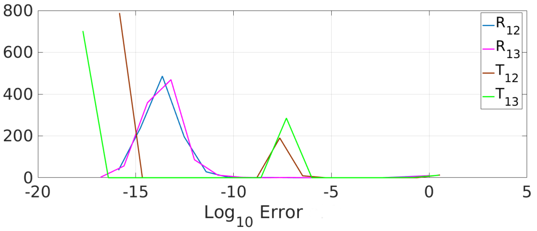

Figure 3. Errors of computed parameters with respect to the ground truth are small showing that the solver is numerically stable for the alternate formulation of Cleveland.

C. Additional Synthetic Experiments

Synthetic experiments were completed for the minor formulation of Cleveland discussed in Section 3.1 in the main manuscript, as well as the other formulations outlined above in Section 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 alternate Chicago and Cleveland, it was found that pose estimation errors are negligible as shown in Figures 1, 2, and 3, respectively.

The next set of experiments show the behavior when the correspondences are correct, but noisy. Using the same process as described in detail in Section 4 of the main paper. The result of the minor formulation of Cleveland and alternate formulations of Chicago and Cleveland are shown in Figures 4, 5, and 6, respectively. For each formulation, the median of the translation and rotation error are low, but due to the relatively high failure rate of these three formulations, there are several failures that effect the data. However, these failure cases can be detected and resolved by thresholding the maximum inlier ratio in RANSAC. In addition, the average reprojection error with respect to the ground truth point correspondences, also in Figures 4, 5, and 6, shows that for most of the test cases we have a stable and reasonable reprojection error. Again, the cases with large reprojection error can be ignored by thresholding maximum inlier ratio in RANSAC.

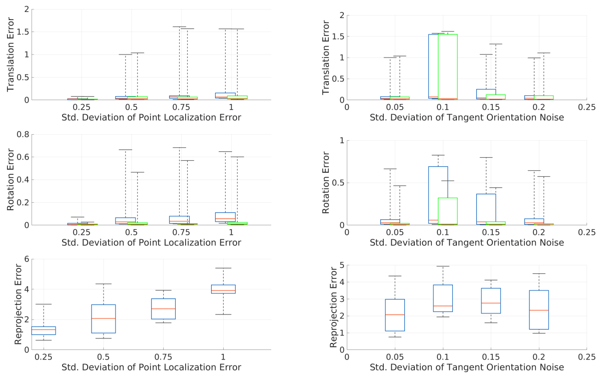

Figure 4. Distribution of trifocal pose error for the minor formulation of Cleveland in the form of translational and rotational error between cameras 1 and 2 (blue) and cameras 1 and 3 (green), as well as the reprojection error, plotted against the level of feature localization noise (left) and orientation noise (right).

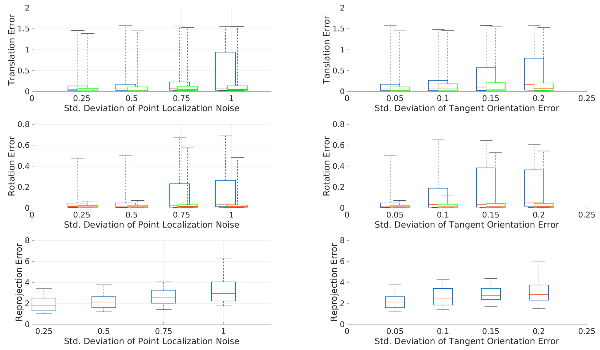

Figure 5. Distribution of trifocal pose error for the alternate formulation of Chicago in the form of translational and rotational error between cameras 1 and 2 (blue) and cameras 1 and 3 (green), as well as the reprojection error, plotted against the level of feature localization noise (left) and orientation noise (right).

These results on synthetic data sets, paired with the computational efficiency of the solvers for various formulations, highlight the efficacy of the homotopy continuation 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 alternate formulation of Chicago, each run takes about 19.69 seconds on average with a failure rate of 12.4% and for the alternate formulation of Cleveland, each run takes about 11.46 seconds on average with a failure rate of 3.2%. All of these tests were done on an AMD Opteron 6378 2.4 GHz processor using 12 threads.

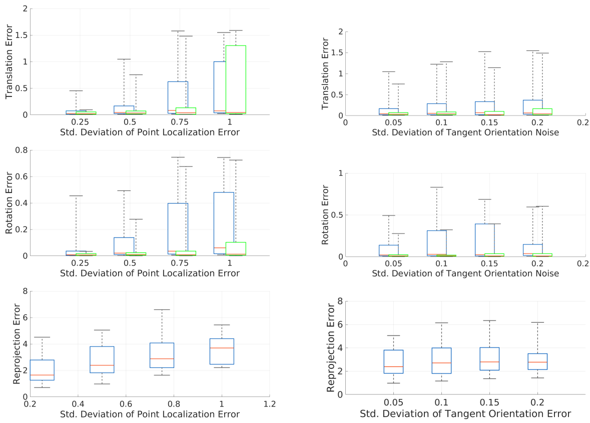

Figure 6. Distribution of trifocal pose error for the alternate formulation of Cleveland in the form of translational and rotational error between cameras 1 and 2 (blue) and cameras 1 and 3 (green), as well as the reprojection error, plotted against the level of feature localization noise (left) and orientation noise (right).

Implementation The minor formulation of Cleveland and the alternate formulations of both Chicago and Cleveland were implemented within a more general purpose solver involving Bertini, which utilizes the parameter homotopy method described in Algorithm 1 in the main paper. They were not implemented in MINUS since this trivial operation would only change speed, and Chicago was the focus of the paper exemplifying this process of transcribing a solver to an optimized C++ version. There are improvements that can be made to precision and error analysis using adaptive multiprecision path tracking [1], 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 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. In fact, in carrying extensive experiments with the synthetic data reported in the paper, after 10000 random solves, the maximum number of iterations for paths leading to ground-truth solutions was 1119, and for the other paths this was 253787. The discrepancy is very large. Given that the solve is about 1 microseconds per iteration, this leads to very good prospects.

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 on reprojection error, the failure rate can be significantly lowered.

The next step for optimizing MINUS is to determine how to prune paths that take a significant length of time to track. 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, would provide strong improvements if ill-conditioning is taken care of. They also indicate that this would improve linear-algebra solves, evaluator lengths, and instruction cache misses. These implementations are currently ongoing.

E. Creation of Mug Dataset

In this work, we created a feature-less mug dataset inspired by Nurutdinova *et al.* [5]. The reason we didn't use the original dataset from [5] is because the occlusion between mug and calibration pattern makes removing the calibration pattern cumbersome. Ten camera poses are set to capture 10 images where the calibration pattern is not occluded. After capturing images, the MATLAB calibration toolbox was used to generate the ground-truth camera pose with manually marked correspondence points on the checkerboard. Once the ground-truth was extracted, the checkerboard area was marked and deleted manually, followed by image inpainting to fill the gap in the image, as shown in Figure 7.

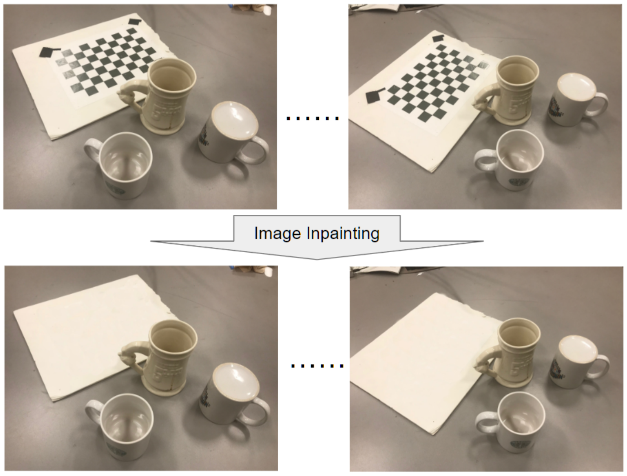

Figure 7. In construction of the mug dataset, a calibration pattern was first placed to generate the ground truth configuration of cameras. Next, the calibration pattern was removed using image inpainting for testing.

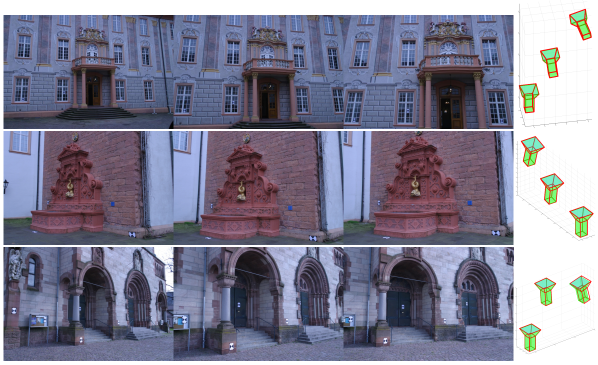

Figure 8. Trifocal relative pose estimation for additional cases from the EPFL dataset. For each row, image triplets samples are shown. The estimation results are shown on the right. Ground truth poses are in solid green and estimated poses are in red.

F. Additional Real Experiments

More real experiments that were not shown in the main paper are shown in this section. First, for texture-rich images, more cases from the EPFL dataset are shown in Figure 8 for the Chicago problem. Second, we include a quantitative comparison to other trifocal methods reported in [4] for the Chicago problem, as shown in Table 1. As in [4], we compare using the two datasets Fountain P-11 and Herz-Jesu-P8, illustrating that our method is comparable to or better than other trifocal methods.

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.

References

[1] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler. Adaptive multiprecision path tracking. *SIAM Journal on Numerical Analysis*, 46(2):722–746, 2008. 3

[2] David Cox, John Little, and Donald O'Shea. *Using Algebraic Geometry*. Springer, 1998. 2

[3] David A. Cox, John Little, and Donald O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2015. 2

[4] Laura Julià and Pascal Monasse. A critical review of the trifocal tensor estimation. In *The Eighth Pacific-Rim Symposium on Image and Video Technology – PSIVT'17*, pages 337–349, Wuhan, China, 2017. Springer. 4

[5] Irina Nurutdinova and Andrew 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*, pages 2363–2371, 2015. 4