Trifocal Relative Pose From Lines at Points

Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret Regan, David da Costa de Pinho, Elias Tsigaridas, Charles Wampler, *Fellow, IEEE*, Jonathan Hauenstein, Peter J. Giblin, Benjamin B. Kimia, Anton Leykin, and Tomas Pajdla, *Member, IEEE*

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 the novel case of (*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 framework MINUS, 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, a key contribution given the borderline intractable degree of nonlinearity of trinocular constraints. 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.

Index Terms—Multiple view geometry, homotopy continuation, structure from motion using curves, numerical algebraic geometry

1 Introduction

Scientific research on 3D reconstruction from multiple views has made an impact [1] by mostly relying on points in Structure from Motion (SfM) [2], [3], [4], [5]. Still, even production-quality SfM technology fails [1] 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.

The failure of bifocal pose estimation using RANSAC on hypothesized correspondences, *e.g.*, using 5 points [6], is highlighted in a dataset of images of mugs, Fig. 1 (similar to the dataset in [7] but without a calibration board), for which the failure rate using the standard SfM pipeline COLMAP [63] is 75%. The failure of just directly applying 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 [2], for robustness. The focus of this paper is to address the failure of traditional bifocal algorithms in such cases, while tackling strategic problems that have long-term potential for breakthrough for a myriad of other minimal problems we jointly discovered and tackled [8], [9], [10], [11], and in the case of curve features for SfM which critically depend on trifocal geometry [12], [13], [14], [15], [16].

The failure of bifocal algorithms motivates the use of (*i*) more complex features, *i.e.*, having additional attributes and (*ii*) more diverse features (partial visibilitiy also helps in robustness, see [17]). We propose that *orientation* (in the sense of inclination) is a key attribute to disambiguate correspondences and we show that SIFT orientation in particular

- *Ricardo Fabbri is with the Department of Computational Modeling, Polytechnic Institute, Rio de Janeiro State University, Nova Friburgo 28625-570, Brazil. E-mail: rfabbri@iprj.uerj.br.*
- *Timothy Duff is with the University of Washington, Seattle, WA 98195-2100 USA. E-mail: tduff3@gatech.edu.*
- *Hongyi Fan and Benjamin B. Kimia are with the School of Engineering, Brown University, Providence, RI 02912 USA. E-mail: hongyi_fan@alumni.brown.edu, kimia@lems.brown.edu.*
- *Margaret Regan is with the Duke University, Durham, NC 27708 USA. E-mail: mregan@math.duke.edu.*
- *David da Costa de Pinho is with the UENF, Campos dos Goytacazes, RJ 28013-602, Brazil. E-mail: davinhopinho@gmail.com.*
- *Elias Tsigaridas is with the INRIA, 75012 Paris, France. E-mail: elias.tsigaridas@inria.fr.*
- *Charles Wampler and Jonathan Hauenstein are with the University of Notre Dame, Notre Dame, IN 46556 USA. E-mail: charles.w.wampler@gm.com, hauenstein@nd.edu.*
- *Peter J. Giblin is with the University of Liverpool, L69 3BX Liverpool, U.K. E-mail: pjgiblin@liverpool.ac.uk.*
- *Anton Leykin is with the Georgia Tech, Atlanta, GA 30332 USA. E-mail: leykin@math.gatech.edu.*
- *Tomas Pajdla is with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, 166 36 Prague, Czechia. E-mail: pajdla@cvut.cz.*

Manuscript received 22 May 2021; revised 1 August 2022; accepted 16 November 2022. Date of publication 0 2022; date of current version 0 2022. NSF DMS-1439786 and the Simons Foundation Grant 507536. Ricardo Fabbri was supported in part by UERJ Prociência award, in part by NSF under Grant IIS-1910530 and in part by FAPERJ Jovem Cientista do Nosso Estado under Grant E-26/201.557/2014. Timothy Duff was supported in part by NSF Mathematical Sciences Postdoctoral Research Fellowship under Grant DMS-2103310, and in part by NSF under Grant DMS-1151297. Benjamin Kimia and Hongyi Fan was supported by the NSF under Grant IIS-1910530. Margaret Regan was supported in part by NSF under Grant CCF-1812746 and in part by the Schmitt Leadership Fellowship in Science and Engineering. Jonathan Hauenstein was supported in part by NSF under Grant CCF-1812746 and in part by ONR under Grant N00014-16-1-2722. Anton Leykin was supported by NSF under Grant DMS-2001267. Tomas Pajdla was supported in part by the EU Regional Development Fund under Grant IMPACT CZ.02.1.01/0.0/0.0/15 003/0000468 and in part by EU H2020 project ARtwin under Grant 856994.
(Corresponding author: Ricardo Fabbri.)
Recommended for acceptance by K. Schindler.
This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2022.3226165, provided by the authors.
Digital Object Identifier no. 10.1109/TPAMI.2022.3226165

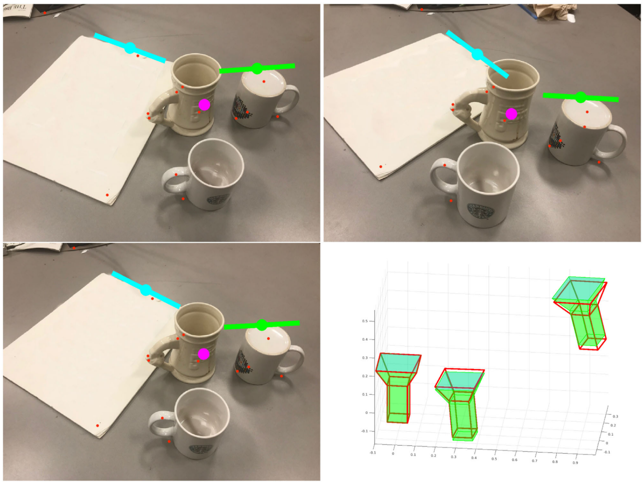

Fig. 1. A deficiency of the traditional two-view approach to bootstraping SfM: not enough features detected (red dots) and thus a SOTASfM pipeline COLMAP fails to reconstruct the relative camera pose. In contrast, the proposed trinocular method requires only three features corresponding in three views: two point-tangents (points with SIFT orientation, green and cyan) and one point without orientation (purple, also SIFT). Red cameras are computed by our approach and green is ground truth.

is a stable feature across views for trifocal pose estimation. Orientation can also arise from curve tangents [14], [15], [18], 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 [12], [13], [14].

Camera estimation from trifocal tensors is long believed to augment two-view pose estimation [19], [20]. Although no significant improvements over bifocal pairwise estimation have been documented [21], recent work reiterate the advantages of well-crafted trifocal algorithms for relevant near-degenerate configurations such as approximately collinear camera centers [20], [22]. The calibrated trinocular relative pose estimation from four points, 3v4p, is notably difficult to solve [15], [23], [24], [25], and is not a minimal problem – it is over-constrained. The first working solver [23] effectively parametrizes the relative pose between two cameras as a curve of degree ten of 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 actual constraints for the 11 unknowns. We refer to this novel problem of three triplets of corresponding points, with two of the points having oriented features as "*Chicago*", which evolved out of the work by Fabbri, Giblin and Kimia on absolute pose estimation from two points endowed with tangents [13], [14]. 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 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 [26], [27], [28] *(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 practical 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) 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 area code of Cleveland). The minimality of combinations of points and lines for the general case [29] 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)* framework MINUS which iteratively tracks solutions with a guarantee of global convergence [27]. Our framework specializes the general HC approach to minimal problems typical of multiple view geometry, dramatically speeding up the implementation. 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: 660 ms 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 [90], as a batch for RANSAC, which may further 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; the provided C++ framework is fully templated to include new minimal problems seamlessly.

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 [30] and Amsterdam Teahouse datasets [31], Figs. 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 [23], [24], [25], [32]. There are no publicly available solvers we are aware of. The state of the art solver [23], based on four corresponding points (3V4P), has not yet found many practical applications [33]. A solver for a relaxed version of this problem has been recently made available by our coauthors based on techniques originated in the present paper [34].

For the projective case, 6 points are needed [35], and Larsson et al. solved the longstanding trifocal minimal problem using 9 lines [36]. The case of mixed points and lines is less common [37], but has seen a growing interest in related problems [38], [39], [40]. The calibrated cases beyond 3V4P are largely unsolved, spurring sophisticated theoretical work [41], [42], [43], [44], [45], [46], [47]. Kileel [43] studied minimal problems in this setting, such as the Cleveland problem solved in the present paper, and reported studies using HC. He stated that the *full* set of ideal generators, *i.e.*, a set of polynomial equations provably necessary and sufficient to describe calibrated trifocal geometry, was unknown.

Seminal works used curves and edges in three views to transfer differential geometry for matching [48], [49], and for pose and trifocal tensor estimation [16], [50], beyond straight lines for uncalibrated [51], [52] and calibrated [38], [53] SfM. Point-tangents – not to be confused with point-rays [54] – 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 curves [55], [56]. Point-tangent fields can be framed as vector fields, so related technology applies to surface-induced correspondence data [15]. In the calibrated setting, point-tangents were first used for absolute pose estimation by Fabbri et. al. [13], [14], from only two points, later relaxed for unknown focal length [57]. The trifocal problem with three point-tangents as a local version of trifocal pose for global curves was first formulated by Fabbri [15], presented here as a minimal version codenamed Chicago.

Homotopy Continuation. The basic theory of polynomial HC [26], [58], [59] 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 [26], [28], [60], [61]. 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 [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], as well as for MRFs [62], [74], and in more recent work [75], [76], [77]. An implementation of the early continuation solver of Kriegman and Ponce [67] by Pollefeys is still widely available for low degree systems [78].

As an early example, HC was used to find an early bound of 600 solutions to trifocal pose with 6 lines [64]. 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. [79] recently compared a general purpose HC solver [61] 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 [27] is more specific than their use of polyhedral homotopy, in the sense that fewer paths are tracked (cf.. the start system hierarchy in [59]).

2 Two Trifocal Minimal Problems

We formulate a new minimal problem for points and incident lines in three views, codenamed *Chicago*. We present its fundamental equations in explicit parametric form that shed light on the geometric properties relevant to vision, as well as a more specific set of equations with 14 unknowns used in our best-performing solver MINUS. While we focus on the Chicago problem, our formulations, analysis and solver framework generalize to important similar problems, and has lead to companion work by our coauthors [29]. To illustrate this, we present a second trifocal problem for points and a free line, codenamed *Cleveland*. The formulation, characterization and practical solver approach for Cleveland, in direct analogy to Chicago, are also a contribution of this paper. Specific details on Cleveland are left for the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3226165, since our focus is on Chicago and the analysis is analogous.

2.1 Formulation and Notation

We follow notational style from Hartley and Zissermann [51] with explicit projective scales. A more elaborate notation [14], [50] can be used to express the equations in terms of tangents to curves and derivatives of relevant quantities such as depth. Fig. 2 illustrates the notation for a single feature consisting of a point and an incident line in three views. Symbols may be given two subscripts $p, v = 1, 2, 3$ to index multiple feature points and views, respectively; indices p may be omitted for brevity.

Let $\mathtt{R}_v, \mathbf{t}_v$ denote the rotation and translation transforming coordinates from camera 1 to camera v (so that $\mathtt{R}_1$ is identity and $\mathbf{t}_1 = 0$). Symbols $\mathbf{X}_p$ and $\mathbf{Y}_p$ denote inhomogeneous coordinates of 3D points, and $\mathbf{x}_{pv}, \mathbf{y}_{pv}$ homogeneous coordinates of their respective projections on $\mathbb{P}^2$ at view v, with α_{pv}, β_{pv} their respective depths. Let $\mathbf{l}_{pv}$ and $\mathbf{L}_p$ denote column vectors of homogeneous coordinates of image lines and underlying 3D lines in $(\mathbb{P}^2)^\vee$ and $(\mathbb{P}^3)^\vee$, resp. We use both parametric and homogeneous equations for lines, the latter

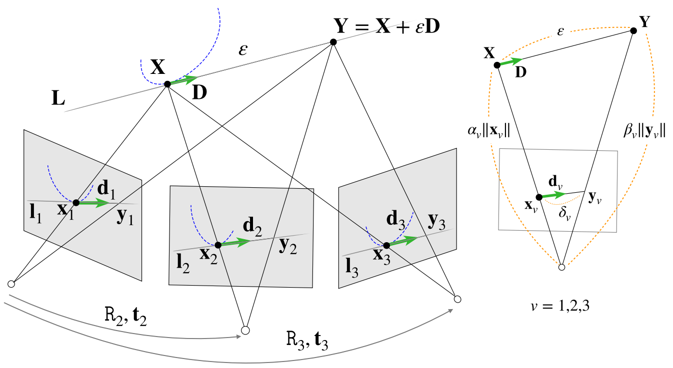

Fig. 2. Notation illustrated for a single point with a curve tangent vector or feature orientation, e.g., SIFT. Multiple features may be explicitly indexed with an additional first subscript.

obtained by eliminating the line parameter from the former. Symbol $\mathbf{d}_{pv}$ represents a line direction or unit curve tangent vector in homogeneous coordinates at view v (point at infinity, i.e., third coordinate is zero); and $\mathbf{D}_p$ is the underlying 3D line direction or space curve tangent in inhomogeneous world coordinates. Displacements ε_p along $\mathbf{D}_p$ correspond to displacements δ_{pv} along $\mathbf{d}_{pv}$. Let $\boldsymbol{\pi}_{pv}$ denote the homogeneous coordinates of the backprojection plane in $(\mathbb{P}^3)^\vee$ of $\mathbf{l}_{pv}$. For simplicity, we use concrete coordinate representations even in coordinate-independent statements. By default, all coordinates are assumed real and without the action of intrinsic parameters.

Definition 1 (Chicago Trifocal Problem). *Given three corresponding points $\mathbf{x}_{1v}, \mathbf{x}_{2v}, \mathbf{x}_{3v}$ and two lines $\mathbf{l}_{1v}$, $\mathbf{l}_{2v}$ in views $v = 1, 2, 3$, such that $\mathbf{l}_{pv}$ meets $\mathbf{x}_{pv}$ for $p = 1, 2$ and $v = 1, 2, 3$, compute relative pose $\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3$.*

Examples of Data for Chicago: 1) Three oriented features (e.g., SIFT) corresponding across three views, using feature orientations; 2) General curves in three views (e.g., linked subpixel edges), and three corresponding curve points (e.g., subpixel edgels), using tangent vectors; 3) Trajectories of three moving points observed by three cameras, using velocity vectors. While a third orientation triplet is usually available and exploited in practice, we show the core pose solution requires only two.

Definition 2 (Cleveland Trifocal Problem). *Given three points x_{1v}, x_{2v}, x_{3v} in views $v = 1, 2, 3$, and given a free line $\mathbf{l}_{1v}$ in each image, compute $\mathtt{R}_2, \mathtt{R}_3, \mathbf{t}_2, \mathbf{t}_3$.*

2.2 Essential Equations

The essential equations of Chicago (and Cleveland) are obtained by writing constraints per feature independently, while keeping the pose unknowns in general form. They are used for analyzing the fundamental properties of the new problems and as a basis for further variable elimination and exploring other formulations. See [80] for a general framework for navigating different formulations. The final solver that offered the best performance uses a formulation that further eliminates variables across these per-feature equations using specific algebraic manipulations connecting features pairwise, as described further in Section 2.3.

Theorem 2.1 (Essential Trifocal Constraints for Points and Incident Lines, Parametric Form). *The constraints on relative pose from points and incident lines observed in three views are given by*

$$\alpha_v \mathbf{x}_v = \mathtt{R}_v \alpha_1 \mathbf{x}_1 + \mathbf{t}_v, \tag{1}$$
$$\eta_v \mathbf{x}_v + \mu_v \mathbf{d}_v = \mathtt{R}_v(\eta_1 \mathbf{x}_1 + \mu_1 \mathbf{d}_1), \tag{2}$$

for $v = 2, 3$ (point indices omitted, $\mathtt{R}_1 = \mathtt{I}$ and $\mathbf{t}_1 = 0$). We call (1) the parametric essential trifocal point constraints, and (2) the parametric essential trifocal incident line constraint. Moreover, (1) imposes three constraints per triplet point, while (2) imposes one constraint per incident line triplet:

1) Point epipolar constraints: *Solving (1) for $v = 2$ and $v = 3$.*
2) Point relative scale constraint: *Enforcing depth α_1 to be equal in (1) for $v = 2$ and $v = 3$.*
3) Incident line constraint: *Jointly expressed by (2) for $v = 2, 3$.*

Proof. Eliminate $\mathbf{X}$ from the projections of points $\alpha_v \mathbf{x}_v = \mathtt{R}_v \mathbf{X} + \mathbf{t}_v$, $v = 1, 2, 3$ to get (1). Lines in space through $\mathbf{X}$ are modeled here in parametric form by a displacement parameter ϵ and points $\mathbf{Y} = \mathbf{X} + \varepsilon \mathbf{D}$, which are projected as $\beta_v \mathbf{y}_v = \mathtt{R}_v \mathbf{Y} + \mathbf{t}_v$, $v = 1, 2, 3$. Eliminate $\mathbf{t}_v$ by subtracting the projection equations of $\mathbf{X}$ and $\mathbf{Y}$, $\beta_v \mathbf{y}_v - \alpha_v \mathbf{x}_v = \varepsilon \mathtt{R}_v \mathbf{D}$, and eliminate $\varepsilon \mathbf{D}$ using the equation for $v = 1$ and $\mathbf{y}_v = \mathbf{x}_v + \delta_v \mathbf{d}_v$

$$(\beta_v - \alpha_v)\mathbf{x}_v + \beta_v \delta_v \mathbf{d}_v = \mathtt{R}_v((\beta_1 - \alpha_1)\mathbf{x}_1 + \beta_1 \delta_1 \mathbf{d}_1), \tag{3}$$

for $v = 2, 3$. We set $\eta_v \doteq \beta_v - \alpha_v$ and $\mu_v \doteq \beta_v \delta_v$, yielding (2).

It follows that the trifocal essential point constraints in parametric form (1) are logically equivalent to the existence of a triangulation $\mathbf{X}$ from views 1 and 2 *equal* that from views 1 and 3. In parametric form, it simply means that these solutions can be linked by the *same* depth α_1. By construction, these imply the existence of a triangulation from views 2 and 3, also equal to $\mathbf{X}$, so (2) for views 2 and 3 does not provide an additional constraint.[1]

The trifocal essential incident line constraints in parametric form are logically equivalent to the existence of a 3D line direction $\mathbf{D}$ that, when rooted at $\mathbf{X}$, projects to direction $\mathbf{d}_1$ and $\mathbf{d}_2$, and that $\mathbf{D}$ also projects to $\mathbf{d}_3$. In the point case the equation from views 1 and 2 provides a constraint, i.e., (1) for $v = 2$ does not always have a solution, while the incident line equation from views 1 and 2 does not provide a constraint on pose – there is always a solution μ and η for (2) for $v = 2$ that parametrizes some consistent $\mathbf{D}$ irrespective of $\mathtt{R}$ and the data $\mathbf{x}$ and $\mathbf{d}$. Each triplet of oriented point features provides a single orientation constraint expressed as two coupled equations (2) in η and μ in addition to pose. □

Corollary 2.2. *The correspondence of points across three views constrain relative rotations and translations, while the additional correspondence of an incident line constrains only rotation.*

Proof. This is a direct consequence of Theorem 2.1. □

1. Conversely, having three pairwise epipolar constraints is *not* equivalent to two pairwise epipolar constraints and a relative scale constraint [22].

Having an incident line thus works like an additional point correspondence – in a precise sense like a third of a point – yet constraining only rotations. This allows us to construct Chicago as an exactly constrained trifocal problem that can be applied, e.g., with conventional SIFT features. We can get an expression of these constraints free of auxiliary parameters by further elimination.

The parametric point epipolar constraints of Theorem 2.1, *in particular*, state that $\mathbf{x}_1$, $\mathbf{x}_v$ and the first camera center $\mathbf{t}_v$ are coplanar when written in the coordinates of camera v; this is the classical Essential constraint, expressed without parameters via a scalar triple product trilinear in $\mathbf{t}_v$ and the points, the standard expression that is bilinear in image coordinates. Although we arrived at this constraint explicitly from first principles through but the simplest logic, it is a general constraint of two-view geometry with recent results in trifocal geometry [20]. Algebraically, the classical expression for the Essential constraint ammounts to eliminating depths α_v from (1) while keeping $\mathtt{R}_v$ and $\mathbf{t}_v$. However, there are successful arguments for eliminating $\mathtt{R}_v$ and $\mathbf{t}_v$ *first* in camera pose problems, writing the equations in terms of depths only α [13], [14] (e.g., the classical P3P equations). Though not performed here, this further motivates stating the trifocal essential constraints in parametric form. Moreover, the parametric form more readily lends itself to modeling general curves [12] for which trifocal geometry plays a pivotal role. The trifocal relative scale constraint in Theorem 2.1 guarantees that 3D rays converge, which may not be the case if we had used three pairwise epipolar constraints instead; in fact, this scale constraint is a fundamental and classical condition of photogrammetry, called the *scale-restraint* equations, see [22] for general results. It may be substituted by an additional epipolar constraint between views 2 and 3, but it turns out that this is only adequate for oriented points, i.e., *together* with the incident line constraint, which guarantees a consistent 3D incident line. Without this, having three pairwise epipolar constraints is not enough to guarantee there is a 3D point $\mathbf{X}$ that projects to the observed points, specially near non-generic configurations [22], namely 1) if the camera centers are far from collinear, when the corresponding rays lie in or near the trifocal plane 2) if the centers are approximately collinear, when the rays lie near any plane containing the baseline [22]. In this sense, points with incident lines are *natural* features in trifocal geometry.

Corollary 2.3 (Chicago Essential Equations, Parametric Form). *The Chicago problem is equivalent to finding the solutions of*

$$\alpha_{pv}\mathbf{x}_{pv} = \mathtt{R}_v\alpha_{p1}\mathbf{x}_{p1} + \mathbf{t}_v, \quad p = 1,2,3 \tag{4}$$

$$\eta_{pv}\mathbf{x}_{pv} + \mu_{pv}\mathbf{d}_{pv} = \mathtt{R}_v\big(\eta_{p1}\mathbf{x}_{p1} + \mu_{p1}\mathbf{d}_{p1}\big), \quad p = 1,2, \tag{5}$$

for $v = 2,3$, which are 30 scalar equations in the relative camera pose R_2, t_2, R_3, t_3, along with 9 unknown depths ($\alpha_p v$) and 12 unknown line parameters (6 each for η_{pv} and μ_{pv}).

Proof. Theorem 2.1 lists all the available constraints. □

That actual equations used in our solver amount to an elimination of the auxiliary parameters in (4) and (5), leading to vanishing minors, Section 2.3. Note that (4) are homogeneous in α and $\mathbf{t}$, so that a multiple of a particular solution are also solutions, i.e., translations and depths are constrained up to scale, giving 11 constrainable degrees of freedom. By Theorem 2.1, the essential equations used in Chicago express 3 independent constraints per point, and 1 per incident line, yielding 11 constraints on 11 degrees of freedom. Rigorous computational arguments in Section 3 confirm that these constraints are also independent across points. In other words, Chicago is a minimal problem.

One can also see the parametric trifocal essential equations for Chicago as a square system of 30 scalar equations in the 30-dimensional space $SO(3) \times SO(3) \times \mathbb{P}^{14} \times \mathbb{P}^5 \times \mathbb{P}^5$ of unknowns

$$(\mathtt{R}_2, \mathtt{R}_3, [\mathbf{t}_2, \mathbf{t}_3, \alpha_{11}, \ldots, \alpha_{33}], [\eta_{11}, \mu_{11}, \eta_{12}, \mu_{12}, \eta_{13}, \mu_{13}], [\eta_{21}, \mu_{21}, \eta_{22}, \mu_{22}, \eta_{23}, \mu_{23}]).$$

We model the 9 depths α_v and $\mathbf{t}_2, \mathbf{t}_3$ as a point in $\mathbb{P}^{14}$, since they are unknown up to a common scale. Since only the directions of tangents matter, we regard these solution components as points in two $\mathbb{P}^5$ factors, one per oriented feature.

There are many ways to proceed with elimination from the essential parametric equations to obtain alternate formulations, as discussed above. A particular eliminated formulation based on vanishing minors, which produced the first working solver for Chicago, and which are used in MINUS, is described in Section 2.3.

2.3 Equations Based on Minors Used in Our Solver

Experiments show that judicious elimination of additional variables from the basic equations leads to faster and more reliable solvers, with tradeoffs, e.g., in the number of variables versus nonlinearity and degeneracy of the resulting representations. This section describes a particular way to eliminate variables down to a 14×14 system that has proven most successful and general to date.

Futher elimination of certain variables from the basic equations leads to minor-based constraints, i.e., enforcing the determinants of certain sub-matrices to vanish. Examples are coplanarity or multilinear constraints, e.g., the essential constraint. In particular, this eliminates parameters describing coordinates of vectors in constraints on lines (depths α's) and planes (η's and μ's). While this approach has long been used for describing trifocal constraints for points [22], in full generality it is novel and has spawned companion work by our coauthors [29]. Additionally, equations based on minors are multilinear, allowing for possible numerical improvements, Section 4.1.

An instance of Chicago may be described by a configuration of 5 *visible lines* in each view, Fig. 3. We denote each line by $\mathbf{l}_{1v}, \ldots, \mathbf{l}_{5v}$ for $v = 1,2,3$, where the first three $\mathbf{l}_{1v}, \mathbf{l}_{2v}, \mathbf{l}_{3v}$ pass through all pairs of points in each view, and the last two $\mathbf{l}_{4v}, \mathbf{l}_{5v}$ represent the point-tangent pairs. The minor-based equations split into three sets summarized as:

Lines correspond: $\boldsymbol{\pi}_{i,1}, \boldsymbol{\pi}_{i,2}, \boldsymbol{\pi}_{i,3}$ meet at a 3D line $\mathbf{L}_i$.
Pairwise lines meet: $\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3$ meet pairwise in 3D.
Incident tangents: $\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_4$ and $\mathbf{L}_1, \mathbf{L}_3, \mathbf{L}_5$ meet at a point.

The latter two are so-called *common point constraints*. *Line correspondence* constraint. These equations express that there must be an underlying 3D line $\mathbf{L}_j$, $j = 1, \ldots, 5$ associated to

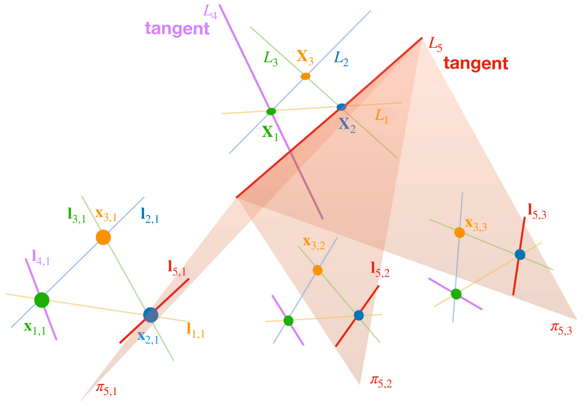

Fig. 3. Visible line diagram for Chicago. Cleveland uses the same numbering for pairwise lines and $\mathbf{l}_4$ is a free line.

the set of backprojection planes $\boldsymbol{\pi}_{j,v} = [\mathtt{R}_v|\mathbf{t}_v]^\top \mathbf{l}_{j,v}$, $v = 1, 2, 3$, which are gathered into a 4×3 matrix $\mathtt{L}_j \doteq [\boldsymbol{\pi}_{j,1} \quad \boldsymbol{\pi}_{j,2} \quad \boldsymbol{\pi}_{j,3}]$. These planes define a single line if the underlying system of equations has a 1D solution, leading to the *rank constraint*

$$\operatorname{rank} \mathtt{L}_j \leq 2, \quad j = 1, \dots, 5. \tag{6}$$

Equivalently, we obtain a polynomial system by setting all 3×3 minors of each $\mathtt{L}_j$ to zero. As explained Section 4, MINUS employs a heuristic to select one such minor per $\mathtt{L}_j$, fixed for given HC starting solutions, yielding *5 final equations* for this constraint. *Pairwise line intersection* constraint. That $\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3$ intersect pairwise can be expressed by

$$\operatorname{rank} [\mathtt{L}_i \quad \mathtt{L}_j] \leq 3, \qquad i < j \in \{1, 2, 3\}, \tag{7}$$

or that all maximal 4×4 minors vanish. We use only $\operatorname{rank} [\mathtt{L}_2\ \mathtt{L}_3] \leq 3$ corresponding to $\mathbf{X}_3$, as the other pairwise intersections will be implicit in the constraint of *incident tangents*. For MINUS we pick only one minor equation for this constraint using the aforementioned heuristics.*Incident tangents* constraint. That tangents intersect at the same point with two other lines can be expressed by forming matrices $\mathtt{X}_1 \doteq [\mathtt{L}_1\ \mathtt{L}_2\ \mathtt{L}_4]$, $\mathtt{X}_2 \doteq [\mathtt{L}_1\ \mathtt{L}_3\ \mathtt{L}_5]$, and requring

$$\operatorname{rank} \mathtt{X}_j \leq 3, \ \ j = 1, 3. \tag{8}$$

All 4×4 minors must vanish, 5 of which are used in MINUS. *The final number of equations consists of 11* fixed, specific vanishing minors. The total number of minors associated with the rank constraints (6),(7),(8) far exceeds the number of unknowns used in our formulation of Chicago. The number of unknowns, as described in the next section, is 14, and the total number of equations implied by these rank constraints is $287 = 5\binom{4}{3} + 2\binom{9}{4} + \binom{6}{4}$. Nevertheless, these 287 equations together with 3 dehomogenization equations (12) will have 312 solutions for *almost all* line configurations encoding an instance of Chicago. In our HC solvers, we work with a 14×14 subsystem of these equations which determine a full-rank submatrix of the 290×14 Jacobian matrix. In this approach, the selection of the actual equations out of a large pool of possibilities is done through computer-assisted heuristics, Section 4. While these general tools aid in understanding the underlying geometry, this becomes concealed. Selecting the appropriate subset of minors, e.g., that ensures the 3D rays for matching points always intersect, is a known problem in the projective case [22]. In that scenario, a different subset of minors may be used depending on *a priori* assumptions on camera configuration (e.g., collinear versus non-collinear camera centers) [20]. An explicit set of vanishing minors for point trifocal geometry and the resulting constraints is studied in a general setting by Trager et. al. [20]. A geometric interpretation is that four minors encode constraints that are trilinear in image coordinates and express that 3D rays must meet at a single point. When 3D rays are viewed from four different appropriate image planes, each vanishing minor may be expressed as requiring three coplanar projected lines meeting at a point [20]. We verify experimentally that our chosen set of minors provides a working solver.

3 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 and orientations in the world frame or camera 1 is a point in the world feature space W. 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 $2, \dots, N$ 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 a viewing map $V : W \times X \to Y$. A camera pose problem is: given $y \in Y$, find $(w, x) \in W \times X$ such that $V(w, x) = y$. The projection $\pi : (w, x) \mapsto x$ 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	$2 \cdot C_L$

Summing all the contributions to $\dim Y - \dim W$, we have

Theorem 3.1. *Let $\langle x \rangle \doteq \max(0, x)$. A necessary condition for a N-camera pose problem to be minimal is*

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

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 $\dim Y > \dim W \times X$.

Definition 4. *The* algebraic degree *of a minimal pose problem is the number of solutions* $(w, x) \in V^{-1}(y)$ *for generic* $y \in Y$.

Both Gröbner bases and HC offer probability-one methods for computing all solutions for a particular problem instance specified by $y \in Y$. Gröbner bases also offer an exact method, when working over $\mathbb{Q}$. However, it is difficult to say when any particular $y \in Y$ will satisfy the necessary genericity conditions to have have this many solutions without knowing the algebraic degree *a priori*. Thus, the following statement has two components: that both problems are minimal (rigorously proven) and that their algebraic degrees are as stated (true with probability one).

Theorem 3.2 (Computational). *The Chicago trifocal problem is minimal with algebraic degree 312, and the Cleveland problem is minimal with algebraic degree 216.*

Proof. To show that a N-camera pose problem is minimal, find $(w, x) \in W \times X$ where the Jacobian of $V(w, x)$ is full rank. For exact values of $(w, x) \in W \times X$ in rational arithmetic, we compute the exact rank of this Jacobian. This proves that the problem is minimal. To compute the algebraic degree of a given problem, we write down a system of polynomial equations in unknowns $(w, x) \in W \times X$ for a randomly chosen y. Since the problem is minimal, we expect that the ideal generated by these polynomials is 0-dimensional. Gröbner bases give standard methods [81] both for checking that this ideal is 0-dimensional and computing its degree. To verify that the degree of the ideal is equal to degree of the minimal problem, we have computed all solutions to the system of polynomials specified by $y \in Y$ and verified that they correspond to valid points $(w, x) \in W \times X$. We carried out this procedure with the minors equations and confirmed the degree using the essential equations and HC. □

Remark. The previous argument depends on the system of equations chosen to model the problem. For instance, if (4),(5) are used, then there exist 312 solutions corresponding to valid points in $W \times X$, plus a small number of degenerate solutions where certain values of the depths α equal zero. Additional polynomial equations which exclude these solutions may be generated using the symbolic technique of *saturation* [81, Sec 4.4]. Such a saturation step is also necessary if rotation matrices are modeled with the quaternion parametrization in (11), since we must rule out degenerate solutions with $w_i^2 + x_i^2 + y_i^2 + z_i^2 = 0$.

A companion work by our coauthors [8] provides Macaulay2 tutorial for the Gröner basis degree proof and other general techniques presented in this section for analyzing Chicago, Cleveland, and a number of related minimal problems using the minors approach. Since Gröbner bases can be used to compute the algebraic degrees of both minimal problems, it is natural to hope that they also can be used to design effective minimal solvers. However, the current leading methods for building minimal solvers (eg. [82], [83], [84]) do not scale well for problems of degree 100 or larger. This is our main motivation for using optimized HC.

4 Optimized Homotopy Continuation Solver

Like other minimal problems in vision, the Cleveland and Chicago problems require us to solve a system of polynomial equations. Crucially, these equations are polynomial in both the input data (points and lines in images) and the unknown quantities to be estimated (cameras and world features.) It is common to call these systems *parametrized polynomial systems*, as the input data parametrize the space of all instances of a given problem. In Section 4.1, we review basic facts about *coefficient parameter homotopy*, a very general framework for solving parametrized polynomial systems based on HC methods. The *parameter homotopies* arising in this framework lie at the core of our HC solvers. To make this general framework concrete, Section 2.3 describes in precise detail one possible strategy for formulating the Cleveland and Chicago problems, in which the depths and displacements are eliminated from the essential equations of Section 2.2. Although these formulations are used in our best-performing solvers to date, we stress that the exact formulation is not essential to the underlying technique. Other formulations of the problem will also give rise to parameter homotopies which can be successfully used within general-purpose software [26], [28] or within our optimized C++ framework MINUS described in Section 4.2.

Acknowledging the promise of further speedups brought by experimenting with different formulations, we observe that our specific parameter homotopies can already be used to solve Chicago and Cleveland in a relatively efficient manner, Section 5. We attribute relatively good run times to two factors. First, the inherent specificity of parameter homotopies when compared to other HC methods; the number of paths to track in a parameter homotopy is precisely the algebraic degree of the problem. Second, we optimize various aspects of HC, such as polynomial evaluation and numerical linear algebra, Section 4.2, along with more aggressive optimization opportunities and tradeoffs.

4.1 Algorithm

We assume that $F(\mathcal{R}; \mathcal{A})$ is a system which is polynomial in both the variables $\mathcal{R}$ and the parameters $\mathcal{A}$. One is interested in efficiently computing the solutions for many instances of the parameters. To compute all nonsingular complex isolated solutions of $F(\mathcal{R}; \mathcal{A}) = 0$ for any given set of target parameters $\mathcal{A}^*$, one may use the parameter homotopy

$$H(\mathcal{R}; s) = F(\mathcal{R}; (1 - s)\mathcal{A}_0 + s\mathcal{A}^*), \tag{9}$$

for $s \in [0, 1)$, Algorithm 4.1. It is assumed that solutions for some starting parameters $\mathcal{A}_0$ have already been computed via some offline, *ab initio* phase, described below, by default hardcoded in MINUS. This initial phase determines representatives of nonsingular isolated solutions, making for faster, more efficient solves for any other parameter values desired, e.g., within RANSAC.

Generically, the homotopy paths are smooth and do not intersect each other. To ensure this (genericity) condition for every homotopy path with probability 1, we may employ the so-called *gamma trick*. This consists in choosing a (random) $\gamma \in \mathbb{C}$ so that the homotopy equation becomes

$$H(\mathcal{R}; s) = F(\mathcal{R}; \phi(s)),$$

where $\phi(s)$ parametrizes an arc, depending on γ, connecting $\mathcal{A}_0$ to $\mathcal{A}^*$ in the parameter space. More explicitly, we define $\phi(s) = (1-\tau(s))\mathcal{A}_0 + \tau(s)\mathcal{A}^*$, with $\tau(s) = \frac{\gamma s}{1+(\gamma-1)s}$, as in Algorithm 4.1. In this way, $\phi(s)$ is a generic path in the complex space without singularities, even if the endpoints are real. However, even though the circular arc depending on γ misses the non-generic points in $\mathbb{C}$ with probability 1, it might happen that the arc is close to these non-generic points; this can cause instability, increase the error or decrease speed in computations. If we run MINUS multiple times with the same data but using different (random) γ's, it results in a dispersion of run times and even occasional failures. The slower running times and the occasional failures happen when γ lands close to certain rays in $\mathbb{C}$ which intersect an appropriately-defined discriminant in the tracking parameter s.

For systems which are linear in the parameters $\mathcal{A}$, it is possible to adapt the gamma trick to work with a simpler *linear segment homotopy*, due to the following calculation

$$\begin{aligned} H(\mathcal{R}; s) &= F(\mathcal{R}; (1-\tau(s))\mathcal{A}_0 + \tau(s)\mathcal{A}^*) \\ &= (1-\tau(s))F(\mathcal{R}; \mathcal{A}_0) + \tau(s)F(\mathcal{R}; \mathcal{A}^*) \\ &= \frac{1}{1+(\gamma-1)s}[(1-s)F(\mathcal{R}; \mathcal{A}_0) + \gamma s F(\mathcal{R}; \mathcal{A}^*)], \end{aligned} \quad (10)$$

where the coefficient $\frac{1}{1+(\gamma-1)s}$ is never zero for real $s \in [0, 1)$ and can be ignored when solving $H(\mathcal{R}; s) = 0$. This one variant of the gamma trick may be preferable to the general one, since it results in cheaper evaluation of the homotopy and its derivatives, and may also lead to better numerical stability.

Minor-based constraints are *multilinear* in the coordinates of each line $\mathbf{l}$ suggesting that a simple variant of the aforementioned "linear" gamma trick will work for related formulations. This will indeed work for Cleveland, where we may treat each coordinate of each line as an independent parameter. However, for Chicago there is an additional subtlety due to the fact that the associated configuration of lines is not general and must satisfy

$$\operatorname{rank}[\mathbf{l}_{1v} \quad \mathbf{l}_{2v} \quad \mathbf{l}_{4v}] \le 2, \quad \operatorname{rank}[\mathbf{l}_{1v} \quad \mathbf{l}_{3v} \quad \mathbf{l}_{5v}] \le 2.$$

For Chicago, treating each coordinate of each line as an independent parameter will not give a valid parameter homotopy; even if $\mathcal{A}$ and $\mathcal{A}^*$ encode valid configurations of lines, points on a circular arc or linear segment connecting them will not. We thus represent tangents

$$\mathbf{l}_{4v} = a_{1v}\mathbf{l}_{1v} + a_{2v}\mathbf{l}_{2v}, \quad \mathbf{l}_{5v} = b_{1v}\mathbf{l}_{1v} + b_{2v}\mathbf{l}_{3v},$$

with 2 independent parameters as a pencil of lines.

A full accounting of the variables and parameters used for Chicago in MINUS is as follows. *14 Variables.* Each translation vector has three unknown components, and the entries of matrices $\mathtt{R}_2$ and $\mathtt{R}_3$ are written as rational homogeneous functions in four unknowns (homogenized Cayley)

$$\mathtt{R}_v = \begin{bmatrix} w_v & -z_v & y_v \\ z_v & w_v & -x_v \\ -y_v & x_v & w_v \end{bmatrix} \begin{bmatrix} w_v & z_v & -y_v \\ -z_v & w_v & x_v \\ y_v & -x_v & w_v \end{bmatrix}^{-1}. \quad (11)$$

56 Parameters. $27 = 3 \times 3 \times 3$ parameters represent three independent lines $\mathbf{l}_{1v}, \mathbf{l}_{2v}, \mathbf{l}_{3v}$ in each view; $12 = 3 \times 2 \times 2$ parameters of the form a_{iv} b_{iv} represent two dependent lines $\mathbf{l}_{4v}, \mathbf{l}_{5v}$ in each view; The remaining $17 = 56 - 39$ parameters consist of $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{C}^5$ and $\mathbf{v}_3 \in \mathbb{C}^7$ which are random coefficients of 3 inhomogeneous linear equations

$$(\mathbf{r}_1 \ 1)\ \mathbf{v}_1 = 0, \quad (\mathbf{r}_2 \ 1)\ \mathbf{v}_2 = 0, \quad (\mathbf{t}_2^\top \ \mathbf{t}_3^\top \ 1)\ \mathbf{v}_3 = 0 \quad (12)$$

that determine affine charts on homogeneous coordinates given by $\mathbf{r}_1 = (w_2, x_2, y_2, z_2)$, $\mathbf{r}_2 = (w_3, x_3, y_3, z_3)$, and $(\mathbf{t}_2^\top \ \mathbf{t}_3^\top)$.

In summary, Chicago may be formulated as a system of 290 equations in 14 variables and 56 parameters. A similar accounting lets us formulate Cleveland as a system of 64 equations in 14 variables and 53 parameters. As previously remarked, we may select a square subsystem F to define the homotopy in (9), provided that the Jacobian $\frac{dF}{d\mathcal{R}}(\mathcal{R}_0; \mathcal{A}_0)$ has full rank for every starting solution $\mathcal{R}_0$. We note that the 276 excess equations need not be algebraic consequences of the 14 that are selected. Nevertheless, the fact that each initial solution $\mathcal{R}_0$ satisfies all 290 equations implies that we do not need to enforce these excess equations explicitly – see, e.g., the discussion in [85, SM Section 16], or the discussion of "side conditions" in [59, Section 7.4].

For Chicago, a precomputed set of 312 starting solutions to the 290×14 system for starting parameters $\mathcal{A}_0$ may be numerically continued to 312 solutions for target parameters $\mathcal{A}$ via (9), where F is a suitable 14×14 square subsystem. To obtain the starting solutions, we first compute a single, random problem-solution pair $(\mathcal{R}_0, \mathcal{A}_0)$, first computing $\mathcal{R}_0$ by fabricating a random scene and cameras, then $\mathcal{A}_0$ by projecting features in each image. From this initial problem-solution pair, we may then generate a complete set of 312 solutions by parameter continuation along random monodromy loops in the space of parameters. Such monodromy-based heuristics are standard in numerical algebraic geometry. A complete description is beyond the scope of this paper, see e.g., [86] or [87], where the latter work describes the implementation we used.

For the minors-based formulation of Chicago, an *ad-hoc* variant of the gamma trick may be be used with the linear segment homotopy (10). The variant is used in the implementation of MINUS, and is based on the following idea: pick $\gamma_1, \gamma_2, \ldots, \gamma_{12}$ at random from the complex unit circle, and consider the parameter values $\mathcal{A}^{\gamma_1,\gamma_2,\ldots\gamma_{12}}$ obtained by the following replacements

$$\begin{array}{lll} \mathbf{l}_{1v} \to \gamma_1\, \mathbf{l}_{1v} & a_{1v} \to \overline{\gamma_1}\, a_{1v} & \\ \mathbf{l}_{2v} \to \gamma_2\, \mathbf{l}_{2v} & a_{2v} \to \overline{\gamma_2}\, a_{2v} & \cdots \end{array} \quad (13)$$

These replacements are designed so that systems parametrized by $\mathcal{A}$ and $\mathcal{A}^{\gamma_1,\gamma_2,\ldots}$ have the same solution sets. Thus, for generic starting and target parameters $\mathcal{A}_0$ and $\mathcal{A}^*$, real or complex, we may numerically continue the solutions of $F(\mathcal{R}; \mathcal{A}_0) = 0$ to those of $F(\mathcal{R}; \mathcal{A}^*) = 0$ using the linear segment connecting $\mathcal{A}_0^{\gamma_1,\gamma_2,\ldots\gamma_{12}}$ and $(\mathcal{A}^*)^{\gamma_1,\gamma_2,\ldots\gamma_{12}}$ in the space of parameters. In our current implementation, these random parameters γ's are sampled independently for start and end systems.

We conclude this section with Algorithm 4.1, which contains a high-level description of our HC solver in pseudocode.

Algorithm 1. Homotopy Continuation Solution Tracker

input: Square 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 $\mathcal{A}^*$; Random $\gamma_1, \gamma_2, \ldots, \gamma_{12} \in \mathbb{C}$
output Set of target solutions $\mathcal{R}^*$ where $F(\mathcal{R}^*;\mathcal{A}^*) = 0$
Setup homotopy
$H(\mathcal{R}; s) = F(\mathcal{R}; (1-s)\mathcal{A}_0^{\gamma_1,\gamma_2,\ldots\gamma_{12}} + s(\mathcal{A}^*)^{\gamma_1,\gamma_2,\ldots\gamma_{12}})$.
for *each start solution* **do**
  $s \longleftarrow 0$;
  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 $H(\mathcal{R}^*, 1) = 0$.

4.2 Implementation

We devised an optimized open source package MINUS – MInimal problem NUmerical Solver, available at github.com/rfabbri/minus. This is an HC framework specialized for minimal problems, templated in C++ enabling efficient specialization for different problems, formulations, and precisions. The most reliable and high-quality solver to date uses a 14×14 minors formulation in double precision (64-bit). The most important optimization is exploiting fixed-length C-style arrays to optimize memory layout for size and locality. We also hardcoded evaluators and used Eigen [88]'s LU decomposition with partial pivoting for linear algebra, which proved accurate as long as double precision is used. The most important compile flag is `-ffast-math`; despite aggressive floating point optimizations, this only affected output within 10^{-10} error.

As shown in Section 5, MINUS runs on average at hundreds of miliseconds and up to $100\times$ faster than general-purpose HC. It can run at a few miliseconds at the cost of reduced success rate in finding the solution, due to more aggressive optimization parameters. Such reduced success rate might be mitigated within RANSAC, if adequately assessed. For instance, we successfully devised a "lossy" HC parameter to constrain the number of predictor iterations per solution path, which have yielded an effective speedup at negligible loss in success rates, Section 5.

The second most important algorithm parameter to vary is the maximum number of correction steps; 4 is the current safe default. Increasing it to 5–7 cuts the runtime down to 280ms. Another is corrector tolerance, which affects how many correction iterations are performed: increasing it $10^4\times$ brings the runtime down to less than 200ms. The error rate for these extreme cases can be as high as 50%, although testing reprojection error to larger practical levels of 1 px precision may bring this figure up.

Like MINUS, widespread fast numerical algorithms to compute simple functions such as `sqrt` solve polynomial equations iteratively, and the key lies in the starting point [89].

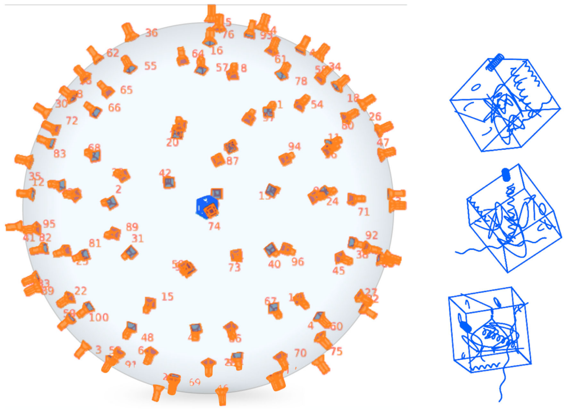

Fig. 4. Sample views of our synthetic dataset. Real datasets have also been used in our experiments. (3D curves are from [12], [14]) .

The start system in MINUS is by default precomputed from random parameters; it could instead be sampled from our synthetic data, and the closest camera could be selected matching a similar configuration of correspondences. See also companion work by our coauthors [34]. Varying the problem formulations also has potential for speedup. Further eliminating variables to, say 6×6, could bring improvements since linear solves could be explicitly inverted. A GPU implementation is explored in companion work by our coauthors [90].

5 Experiments

Experiments are conducted first for synthetic data for a controlled study, followed by challenging real data. We present results for the more challenging Chicago problem, since the exact same core solver is used for Cleveland.

Synthetic data experiments: The synthetic data from [12], [14] consists of 3D curves in a $4 \times 4 \times 4\,\mathrm{cm}^3$ volume projected to 100 cameras (Fig. 4), and sampled to get 5117 points endowed with orientations (tangents of curves) that are projections of the same 3D analytic points and tangents, and then degraded with noise and outliers. Camera centers are randomly sampled on an average sphere around the scene along normally distributed *radii* of mean 1 m and $\sigma = 10$ mm. Rotations are constructed via normally distributed look-at directions with mean along the sphere radius looking to the object, and $\sigma = 0.01$ rad such that the scene does not leave the viewport, followed by uniformly distributed roll. This sampling is filtered such that no two cameras are within 15° of each other. Each camera encompasses a 500×500 px viewport, where the entire dataset is visible at sub-pixel precision with no more than one sample per pixel.

Our first experiment studies the numerical stability of the MINUS solver. The dataset provides veridical 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. Only real solutions that generate positive depth are retained. The unused tangent of the third triplet is used to verify the solution as it provides an unused equation. For each of the remaining solutions only one pose is determined.

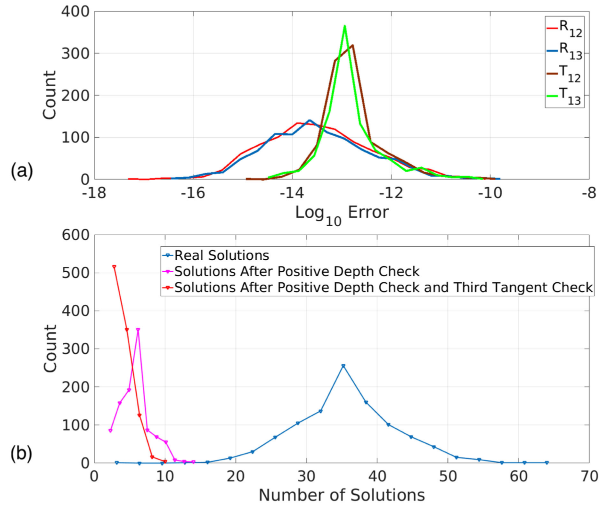

Fig. 5. (a) Errors of computed pose are small showing that the solver is numerically stable. (b) The distributions of the numbers of solutions.

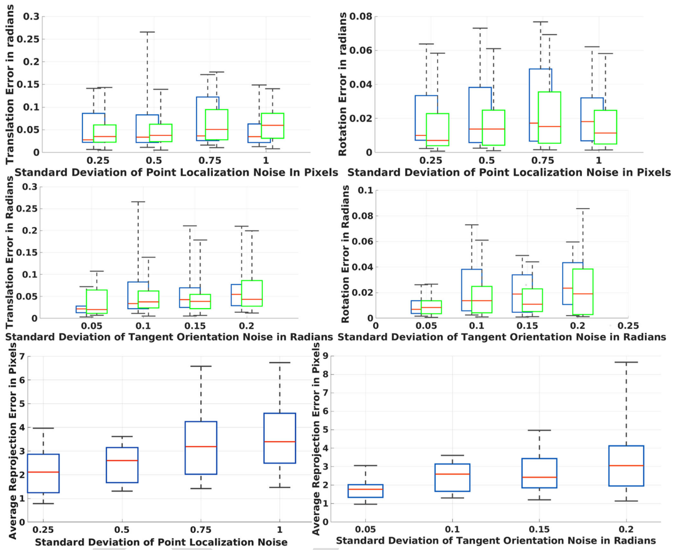

Fig. 6. Pose error between views 1 and 2 (blue) and 1 and 3 (green) versus feature localization (top) and orientation noise (middle), and point reprojection error versus localization and orientation noise (bottom).

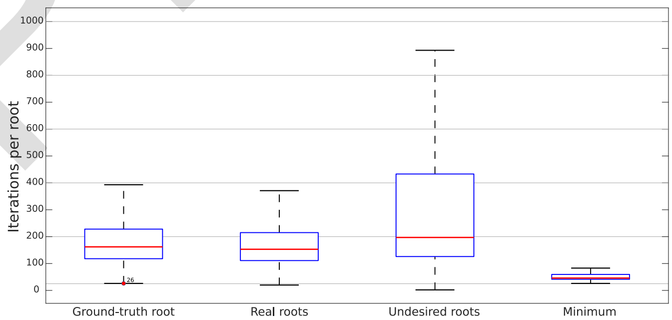

Fig. 7. Time (1 iteration $\approx 1\,\mu s$) spent in root paths leading to ground-truth versus real and undesired roots is stable across 140 generic per configuration. The distribution of the minimum number of iterations to find a root (right) among 1000 randomizations shows the approach can run at the microsecond scale.

The error in pose estimation is the angular error between the normalized translation vectors and between the quaternions. The process of generating the input to pose computation is repeated 10^3 times and averaged. This experiment demonstrates that: (*i*) pose estimation errors are negligible, Fig. 5a; (*ii*) the number of actual 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, Fig. 5b; these extra solutions can be detected by RANSAC; (*iii*) the solver fails in about 1% of cases, which, 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 computational cost.

The second experiment shows that we can reliably and accurately determine camera pose with correct but noisy data. 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 and to the orientation of the tangents, reflecting expected feature localization and orientation localization error. The noise levels on points and tangents reflect those found in curve extraction methods [91]. A RANSAC scheme determines the feature set that generates the most inliers. Experiments indicate that the translation and rotation errors are reasonable. Fig. 6 (top) shows how localization error affects pose under a fixed orientation perturbation of $0.1\mathrm{rad}$; Fig. 6 (bottom) shows how the extent of orientation error affects pose under a fixed localization error of $0.5\mathrm{px}$. The reprojection error, *i.e.*, the distance of a point from the location determined by the other two, is shown in Fig. 6 (bottom), averaged over 100 triplets.

The third experiment shows the system can consistently estimate trifocal pose in the presence of outliers. With a feature localization error of 0.25 px and orientation error of 0.1 rad, 200 triplets of features were generated, with a fraction having random location and orientation. The ratio of outliers is varied over 10%, 25% and 40%, with the experiment repeated 100 times each. The resulting reprojection error is small and extremely stable, with median 2 px and maximum 3.6 px for all outlier ratios.

Computational Efficiency. Each solve with conservative parameters takes 440 ms (660 ms in the worst case), compared to over 1 minute on average for general purpose HC software [26], [28], on an Intel core i7-7920HQ with processor, GCC 5, and four threads. More aggressive potentially unsafe optimizations towards microseconds are feasible, but require assessing failure rate.

To assess putting a cap N_{max} on the number of predictor iterations per root, we first observe that after 10^4 random solves on synthetic data, the maximum number of iterations for paths leading to ground-truth was close to 10^3, versus about 254×10^3 for the wasted paths. Given that the solve is $\approx 1-4\,\mu s$ per iteration, this leads to concrete routes to optimization. Fig. 7 shows that the time for roots leading to ground truth versus undesired paths differ but remain strikingly stable across 140 different random input configurations. For each configuration out of 140, MINUS was run 500 times with different randomizations to find the ground truth parameters. The minimum

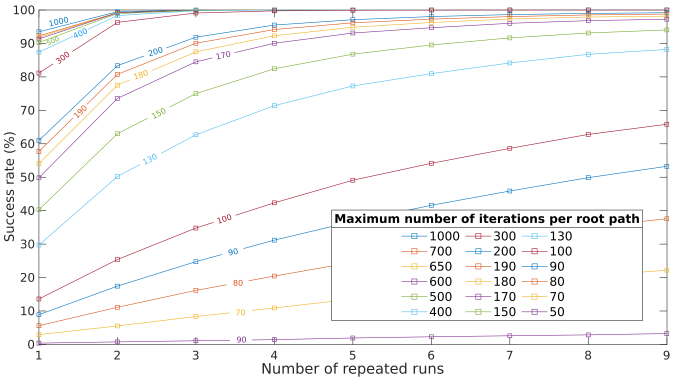

Fig. 8. Tradeoff of success rate versus number of iterations per root.

Fig. 9. Trifocal relative pose results for EPFL dataset. Each row shows images with ground truth (green) and estimated poses (red outline).

number of iterations for all configurations was 26 being consistently less than 100.

Setting $N_{max} < 10^3$ costs a decrease in success rate, Fig. 8. However, we can regain success rate by re-runing N_{rep} times with different randomizations. Fig. 8 shows that running once with $N_{max} = 500$ yields a success of 92%, which is the current default for MINUS, providing the average figure of $401\ ms$. Running thrice with $N_{max} = 200$ yields a similar success rate. For each (N_{max}, N_{rep}) operating point, a success is counted if MINUS found the solution in *any* N_{rep} runs; the final success rate is averaged by performing this procedure 7000 times. If all points have tangents, e.g., 3 SIFT features, as soon as a root reached an HC stop condition we test for positive depth and stop upon compliance with the third tangent to produce a hypothesis for RANSAC, cutting down average execution time further with a modest decrease in success rate. The run time remains on the order of 100 ms.

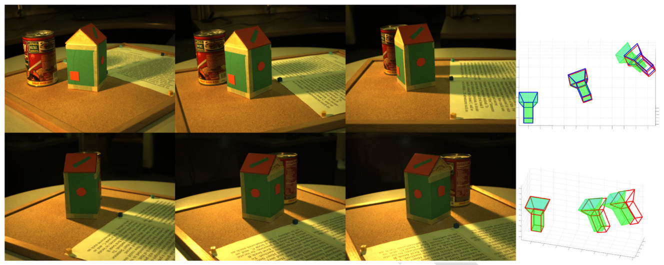

Fig. 10. Trifocal relative pose results for Amsterdam Teahouse: a triplet of images that COLMAP is able to tackle (top) and where it fails (bottom). Results: COLMAP (blue outline), ours (red), and ground truth (green).

Fig. 11. Trifocal relative pose results for a dataset comprising three mugs, which is challenging for traditional SfM. Each shows images with ground truth (green) and estimated poses (red outline).

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 SIFT orientations and a third point without orientation.

Fig. 9 shows that camera pose is reliably and accurately found using triplets of images from the EPFL dense multi-view stereo image dataset [30]. Our quantitative estimates on 150 random triplets from this dataset give pose errors of 1.5×10^{-3} rad in translation and 3.24×10^{-4}rad in rotation. The average reprojection error is 0.31 px. These are comparable to or better than the interest point-based trifocal relative pose estimation methods reported in [21]. Our conclusion for this dataset, whose purpose is to validate the solver, is that our method is at least as good and often better than the traditional ones. 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 10^3 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 [31], which also has ground-truth relative pose data, depicts scenes with

TABLE 1
Pose Error of Our Method Versus 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**

Our method has better rotation error and comparable translation error.

fewer features. Fig. 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 [3]). However, when the number of features is reduced, as in Fig. 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.

We also created another featureless dataset similar to the one in [7] 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 Figs. 1 and 11.

A quantitative comparison with the trifocal methods reported in [21] on datasets Fountain P-11 and Herz-Jesu-P8 is shown in Table 1 for Chicago, illustrating that our method is comparable to or better than other trifocal methods.

6 Conclusion

We presented a new calibrated trifocal minimal problem, an analysis demonstrating its number of solutions, and a practical solver by specializing computation techniques from numerical algebraic geometry. We showed our approach generalizes to characterize and solve a similar difficult minimal problem with mixed points and lines in three views. Both problems are representative of a myriad of similar minimal problems in multiple views analyzed with the techniques initiated with the present work [8], [9], [10], [11], [29]. The increased ability to solve trifocal problems with points and lines is key to future work on broader problems appearing when observing general 3D curves, e.g., in scenes without enough point features, using differential geometry [12], [15]. As a first step, our trifocal solvers have been partially integrated into the SfM pipeline OpenMVG [92] for use with SIFT orientation, and we are working to integrate and verify their robustness advantages also with COLMAP. Our "100 lines of custom-made solution tracking code" have also already been employed to build practical, fast solvers [34] for other minimal problems which have not been efficiently solved with Gröbner bases [84].

Acknowledgments

The authors would like to thank Juliana Santos Barcellos Chagas Ventura for helping with performance experiments, Figs. 7 and 8. This work initiated while the 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, with early versions at arxiv 23Mar2019 04:26UTC and CVPR 2020.

References

[1] ARKit Team, "Understanding ARKit tracking and detection," Apple, WWDC, 2018. [Online]. Available: developer.apple.com/videos/play/wwdc2018/610
[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 72–79.
[3] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.
[4] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.
[5] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 652–659.
[6] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.
[7] I. Nurutdinova and A. Fitzgibbon, "Towards pointless structure from motion: 3D reconstruction and camera parameters from general 3D curves," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2363–2371.
[8] T. Duff, K. Kohn, A. Leykin, and T. Pajdla, "PLMP - point-line minimal problems in complete multi-view visibility," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1675–1684.
[9] T. Duff, V. Korotynskiy, T. Pajdla, and M. H. Regan, "Galois/monodromy groups for decomposing minimal problems in 3D reconstruction," 2021, *arXiv 2105.04460*.
[10] R. Fabbri, "Algebraic computer vision advances the 3D reconstruction of curves and surfaces from multiple views," *Res. Highlight, NSF Math. Sci. Inst.*, May 2020. [Online]. Available: mathinstitutes.org/highlights/algebraic-computer-vision-advances-the-3d-reconstruction-of-curves-and-surfaces-from-multiple-views
[11] B. Kimia, "A differential geometry paradigm for constructing a semantic mid-level representation for multinocular pose estimation and reconstruction," *NSF Grant*, Aug. 2019-2022. [Online]. Available:www.nsf.gov/awardsearch/showAward?AWD_ID=1910530
[12] R. Fabbri and B. B. Kimia, "Multiview differential geometry of curves," *Int. J. Comput. Vis.*, vol. 117, pp. 1–23, 2016.
[13] R. Fabbri, P. J. Giblin, and B. B. Kimia, "Camera pose estimation using first-order curve differential geometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3321–3332, Oct. 2021.
[14] R. Fabbri, P. J. Giblin, and B. B. Kimia, "Camera pose estimation using first-order curve differential geometry," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2012, pp. 231–244.
[15] R. Fabbri, "Multiview differential geometry in application to computer vision," Ph.D. Dissertation, Division of Engineering, Brown University, Providence, RI, USA, Jul. 2010.
[16] C. Schmid and A. Zisserman, "The geometry and matching of lines and curves over multiple views," *Int. J. Comput. Vis.*, vol. 40, no. 3, pp. 199–233, 2000.
[17] F. Vasconcelos, J. P. Barreto, and E. Boyer, "Automatic camera calibration using multiple sets of pairwise correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 791–803, Apr. 2018.
[18] D. Barath and Z. Kukelova, "Homography from two orientation- and scale-covariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1091–1099.
[19] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*. Cambridge, MA, USA: MIT Press, 2001.
[20] M. Trager, M. Hebert, and J. Ponce, "Coordinate-free Carlsson-Weinshall duality and relative multi-view geometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 225–233.

[21] L. Julià and P. Monasse, “A critical review of the trifocal tensor estimation,” in *Proc. 8th Pacific-Rim Symp. Image Video Technol.*, 2017, pp. 337–349.

[22] M. Trager, J. Ponce, and M. Hebert, “Trinocular geometry revisited,” *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 134–152, 2016.

[23] D. Nistér and F. Schaffalitzky, “Four points in two or three calibrated views: Theory and practice,” *Int. J. Comput. Vis.*, vol. 67, no. 2, pp. 211–231, 2006.

[24] L. Quan, B. Triggs, and B. Mourrain, “Some results on minimal Euclidean reconstruction from four points,” *J. Math. Imag. Vis.*, vol. 24, no. 3, pp. 341–348, 2006.

[25] L. Quan, B. Triggs, B. Mourrain, and A. Ameller, “Uniqueness of minimal Euclidean reconstruction from 4 points,” INRIA, Tech. Rep. inria-00548269, 2001.

[26] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, *Numerically Solving Polynomial Systems With Bertini*, vol. 25, Philadelphia, PA, USA: SIAM, 2013.

[27] T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, and J. Sommars, “Solving polynomial systems via homotopy continuation and monodromy,” *IMA J. Numer. Anal.*, vol. 39, no. 3, pp. 1421–1446, 2018.

[28] A. Leykin, “Numerical algebraic geometry,” *J. Softw. Algebra. Geometry*, vol. 3, pp. 5–10, 2011.

[29] T. Duff, K. Kohn, A. Leykin, and T. Pajdla, “PLMP-point-line minimal problems in complete multi-view visibility,” 2019, *arXiv:1903.10008*.

[30] 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 *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[31] A. Usumezbas, R. Fabbri, and B. B. Kimia, “From multiview image curves to 3D drawings,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 70–87.

[32] V. Rodehorst, “Evaluation of the metric trifocal tensor for relative three-view orientation,” in *Proc. Int. Conf. Appl. Comput. Sci. Math. Architect. Civil Eng.*, 2015, pp. 166–172.

[33] Y. Kuang, M. Oskarsson, and K. Åström, “Revisiting trifocal tensor estimation using lines,” in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 2419–2423.

[34] P. Hruby, T. Duff, A. Leykin, and T. Pajdla, “Learning to solve hard minimal problems,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5532–5542.

[35] A. Heyden, “Reconstruction from image sequences by means of relative depths,” in *Proc. 5th Int. Conf. Comput. Vis.*, 1995, pp. 1058–1063.

[36] V. Larsson, K. Åström, and M. Oskarsson, “Efficient solvers for minimal problems by syzygy-based reduction,” in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 2383–2392.

[37] M. Oskarsson, A. Zisserman, and K. Astrom, “Minimal projective reconstruction for combinations of points and lines in three views,” *Image Vis. Comput.*, vol. 22, no. 10, pp. 777–785, 2004, 2002.

[38] Y. Salaün, R. Marlet, and P. Monasse, “Robust and accurate line- and/or point-based pose estimation without manhattan assumptions,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 801–818.

[39] A. Qadir and J. Neubert, “A line-point unified solution to relative camera pose estimation,” 2017, *arXiv:1710.06495*.

[40] A. Vakhitov, V. Lempitsky, and Y. Zheng, “Stereo relative pose from line and point feature triplets,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 662–677.

[41] C. Aholt and L. Oeding, “The ideal of the trifocal variety,” *Math. Comput.*, vol. 83, pp. 2553–2574, 2014.

[42] A. Alzati and A. Tortora, “A geometric approach to the trifocal tensor,” *J. Math. Imag. Vis.*, vol. 38, no. 3, pp. 159–170, Nov. 2010.

[43] J. Kileel, “Minimal problems for the calibrated trifocal variety,” *SIAM J. Appl. Algebra Geometry*, vol. 1, no. 1, pp. 575–598, 2017.

[44] S. Leonardos, R. Tron, and K. Daniilidis, “A metric parametrization for trifocal tensors with non-colinear pinholes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 259–267.

[45] E. Martyushev, “On some properties of calibrated trifocal tensors,” *J. Math. Imag. Vis.*, vol. 58, no. 2, pp. 321–332, 2017.

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

[47] L. Oeding, “The quadrifocal variety,” 2015, *arXiv:1501.01266*.

[48] N. Ayache and L. Lustman, “Fast and reliable passive trinocular stereovision,” in *Proc. 1st Int. Conf. Comput. Vis.*, 1987, pp. 422–427.

[49] L. Robert and O. D. Faugeras, “Curve-based stereo: Figural continuity and curvature,” in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 1991, pp. 57–62.

[50] R. Cipolla and P. Giblin, *Visual Motion of Curves and Surfaces*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[51] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge, U.K.: Cambridge Univ. Press, 2004.

[52] A. Bartoli and P. Sturm, “Structure-from-motion using lines: Representation, triangulation, and bundle adjustment,” *Comput. Vis. Image Understanding*, vol. 100, no. 3, pp. 416–441, 2005.

[53] Y. Salaün, R. Marlet, and P. Monasse, “Line-based robust SfM with little image overlap,” in *Proc. IEEE Int. Conf. 3D Vis.*, 2017, pp. 195–204.

[54] F. Camposeco, T. Sattler, and M. Pollefeys, “Minimal solvers for generalized pose and scale estimation from two rays and one point,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 202–218.

[55] B. Johansson, M. Oskarsson, and K. Astrom, “Structure and motion estimation from complex features in three views,” in *Proc. Indian Conf. Comput. Vision, Graph., Image Process.*, 2002.

[56] J. Zhao, L. Kneip, Y. He, and J. Ma, “Minimal case relative pose computation using ray-point-ray features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1176–1190, May 2020.

[57] Y. Kuang and K. Åström, “Pose estimation with unknown focal length using points, directions and lines,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 529–536.

[58] A. Morgan, *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, Philadelphia, PA, USA: SIAM, 2009.

[59] A. J. Sommese and C. W. Wampler II, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. Hackensack, NJ, USA: World Scientific Publishing Co Pte. Ltd., 2005.

[60] 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 *Proc. Int. Congr. Math. Softw.*, 2014, pp. 183–190.

[61] J. Verschelde, “Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation,” *ACM Trans. Math. Softw.*, vol. 25, no. 2, pp. 251–276, Jun. 1999.

[62] A. M. Bruckstein, R. J. Holt, and A. N. Netravali, “How to catch a crook,” *J. Vis. Commun. Image Representation*, vol. 5, no. 3, pp. 273–281, 1994.

[63] O. D. Faugeras, Q. T. Luong, and S. J. Maybank, “Camera self-calibration: Theory and experiments,” in *Proc. Eur. Conf. Comput. Vis.*, 1992, pp. 321–334.

[64] R. J. Holt and A. N. Netravali, “Motion and structure from line correspondences: Some further results,” *Int. J. Imag. Syst. Technol.*, vol. 5, no. 1, pp. 52–61, 1994.

[65] R. J. Holt and A. N. Netravali, “Number of solutions for motion and structure from multiple frame correspondence,” *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 5–15, May 1997.

[66] R. J. Holt, A. N. Netravali, and T. S. Huang, “Experience in using homotopy methods to solve motion estimation problems,” in *Curves and Surfaces in Computer Vision and Graphics*, L. A. Ferrari and R. J. P. de Figueiredo Eds., Bellingham, WA, USA: SPIE, 1990, pp. 219–226.

[67] D. J. Kriegman and J. Ponce, “A new curve tracing algorithm and some applications,” in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker Eds., San Diego, CA, USA: Academic, 1991, pp. 267–270.

[68] D. J. Kriegman and J. Ponce, “Geometric modeling for computer vision,” in *Curves and Surfaces in Computer Vision and Graphics II*, M. J. Silbermann and H. D. Tagare Eds., Bellingham, WA, USA: SPIE, 1992, pp. 250–260.

[69] Q.-T. Luong, “Matrice fondamentale et calibration visuelle sur l’environnement-vers une plus grande autonomie des systemes robotiques,” Ph.D. dissertation, Paris XI - Automatique/Robotique, Université de Paris-Sud, Centre d’Orsay, 1992.

[70] S. J. Maybank and O. D. Faugeras, “A theory of self-calibration of a moving camera,” *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 123–151, 1992.

[71] S. Petitjean, “Algebraic geometry and computer vision: Polynomial systems, real and complex roots,” *J. Math. Imag. Vis.*, vol. 10, no. 3, pp. 191–220, May 1999.

[72] S. Petitjean, J. Ponce, and D. J. Kriegman, “Computing exact aspect graphs of curved objects: Algebraic surfaces,” *Int. J. Comput. Vis.*, vol. 9, no. 3, pp. 231–255, Dec. 1992.

[73] M. Pollefeys and L. Van Gool, “Stratified self-calibration with the modulus constraint,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 707–724, Aug. 1999.

[74] P. K. Nanda, U. B. Desai, and P. G. Poonacha, “A homotopy continuation method for parameter estimation in MRF models and image restoration,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 1994, pp. 273–276.

[75] A. Ecker and A. D. Jepson, "Polynomial shape from shading," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 145–152.
[76] J. D. Hauenstein and M. H. Regan, "Adaptive strategies for solving parameterized systems using homotopy continuation," *Appl. Math. Comput.*, vol. 332, pp. 19–34, 2018.
[77] M. Salzmann, "Continuous inference in graphical models with polynomial energies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1744–1751.
[78] M. Pollefeys, "VNL RealNPoly: A solver to compute all the roots of a system of n polynomials in n variables through continuation," 1997. [Online]. Available: github.com/vxl,core/vnl/algo/vnl_rnpoly_solve.h
[79] Y. Kasten, M. Galun, and R. Basri, "Resultant based incremental recovery of camera pose from pairwise matches," 2019, *arXiv:1901.09364*.
[80] S. Agarwal, T. Duff, M. Lieblich, and R. Thomas, "An atlas for the pinhole camera," 2022, *arXiv:2206.13468*.
[81] D. A. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Berlin, Germany: Springer, 2015.
[82] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 302–315.
[83] V. Larsson, K. Åström, and M. Oskarsson, "Efficient solvers for minimal problems by syzygy-based reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*,2017, 2017, pp. 2383–2392.
[84] V. Larsson, M. Oskarsson, K. Åström, A. Wallis, Z. Kukelova, and T. Pajdla, "Beyond grobner bases: Basis selection for minimal solvers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3945–3954.
[85] P. Hruby, T. Duff, A. Leykin, and T. Pajdla, "Learning to solve hard minimal problems," 2021, *arXiv:2112.03424*.
[86] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler, "Adaptive multiprecision path tracking," *SIAM J. Numer. Anal.*, vol. 46, no. 2, pp. 722–746, 2008.
[87] T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, and J. Sommars, "Solving polynomial systems via homotopy continuation and monodromy," *IMA J. Numer. Anal.*, vol. 39, no. 3, pp. 1421–1446, 2018.
[88] G. Guennebaud et al., Eigen, 2022. [Online]. Available: http://eigen.tuxfamily.org
[89] L. Parrilla, A. Lloris, A. Castillo, and E. Garcia, "Table-free seed generation for hardware newton-raphson square root and inverse square root implementations in IoT devices," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6985–6995, May 2022.
[90] C.-H. Chien, H. Fan, A. Abdelfattah, E. Tsigaridas, S. Tomov, and B. Kimia, "GPU-based homotopy continuation for minimal problems in computer vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15765–15776.
[91] B. B. Kimia, X. Li, Y. Guo, and A. Tamrakar, "Differential geometry in edge detection: Accurate estimation of position, orientation and curvature," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1573–1586, Jul. 2019.
[92] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *Proc. Int. Workshop Reproducible Res. Pattern Recognit.*, 2016, pp. 60–74.

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

Timothy Duff received the PhD degree in algorithms, combinatorics, and optimization from Georgia Tech, supervised by Anton Leykin. He is an NSF Mathematical Sciences Postdoctoral Research fellow and postdoctoral scholar with the University of Washington. He is a member of SIAM and the AMS.

Hongyi Fan received the MSc degree in computer engineering from Brown University, in 2016. He is currently working toward the PhD degree from Brown University. His research interests include computer vision, 3D reconstruction, minimal problems and their application.

Margaret Regan received the PhD degree in applied mathematics from the University of Notre Dame, USA, in 2020. She is currently an Elliott assistant research professor with the Department of Mathematics, Duke University. Her research interests include numerical algebraic geometry, commutative and homological algebra, and their applications to various fields within science and engineering. She is a member of SIAM and the AMS.

David da Costa de Pinho received the undergraduate degree in mathematics, the MSc degree in computational modeling and 3D computer vision from the Polytechnic Institute, Universidad del Estado de Rí o de Janeiro, Brazil, and the PhD degree in oil exploration engineering from Universidade Estadual do Norte Fluminense Darcy Ribeiro, Brazil, in 2020. He is a professor with Fluminense Federal Institute.

Elias Tsigaridas received the PhD degree from the National Kapodistrian University of Athens, Greece, in 2006. Since 2012 is a permanent senior scientist with Inria Paris; since 2020 he also holds a part-time teaching position with École Polytechnique. He research interests lie at the intersection of computational algebra and geometry and their application in science and engineering.

Charles W. Wampler (Fellow, IEEE) received the PhD degree in mechanical engineering from Stanford University. He is a sr. technical fellow with the General Motors R&D Center, Warren, Michigan, USA, where he has been employed since 1985, currently as a member of the Battery Cell Systems Research Lab. He is a Fellow of ASME, and SIAM and is a member of the National Academy of Engineering.

Jonathan D. Hauenstein received the PhD degree in mathematics from the University of Notre Dame. He held positions with the Fields Institute, Texas A&M University, Mittag-Leffler Institute, North Carolina State University, ICERM and the Simons Institute before returning to the University of Notre Dame as a professor, in 2014. His research includes numerical algebraic geometry and its applications involving a variety of fields in science and engineering.

Peter J. Giblin is professor of mathematics emeritus with the University of Liverpool, U.K. His current research interests are in applications of singularity theory to problems of differential geometry. He also works with local high schools in supervising students in advanced project work in mathematics. In the queen's birthday honours 2018 he was awarded an OBE for services to mathematics.

Benjamin B. Kimia received the PhD in electrical and computer engineering from McGill University. He is professor of engineering with Brown Unviersity. His research includes computer vision and medical imaging inspired by neurophysiology and psychophysics. His expertise includes the representation of shape in 2D, 3D and multiview reconstruction, applied to large image databases, archaeology, assistance for the blind, odometry, and image-guided treatments.

Anton Leykin received the PhD degree from the University of Minnesota, Twin Cities. He works in nonlinear algebra with a view towards algorithms and applications. A large part of his recent work concerns homotopy continuation methods, which includes both theory and implementation in Macaulay2 computer algebra system. He is a member of the ACM, AMS, and SIAM. Google Scholar: scholar.google.com/citations?user=ztNR6w8AAAAJ

Tomas Pajdla (Member, IEEE) received the MSc and PhD degrees from the Czech Technical University, Prague. He works in geometry and algebra of computer vision and robotics, with emphasis on non-classical cameras, 3D reconstruction, and industrial vision. He contributed to the epipolar geometry of panoramic cameras, non-central camera models, generalized epipolar geometries, and to developing solvers for minimal problems in structure from motion.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.