PLMP – Point-Line Minimal Problems in Complete Multi-View Visibility

Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomas Pajdla, *Member, IEEE*

***Abstract*—We present a complete classification of all minimal problems for generic arrangements of points and lines completely observed by calibrated perspective cameras. We show that there are only 30 minimal problems in total, no problems exist for more than 6 cameras, for more than 5 points, and for more than 6 lines. We present a sequence of tests for detecting minimality starting with counting degrees of freedom and ending with full symbolic and numeric verification of representative examples. For all minimal problems discovered, we present their algebraic degrees, i.e.the number of solutions, which measure their intrinsic difficulty. It shows how exactly the difficulty of problems grows with the number of views. Importantly, several new minimal problems have small degrees that might be practical in image matching and 3D reconstruction.**

***Index Terms*—3D reconstruction, minimal problems, multi-view geometry.**

I. Introduction

MINIMAL problems [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29] play an important role in 3D reconstruction [30], [31], [32], image matching [33], visual odometry [34], [35] and visual localization [36], [37], [38]. Many minimal problems have been described and solved, and new minimal problems are constantly appearing. In this paper, we present a step towards a complete characterization of all minimal problems for points, lines, and their incidences in calibrated multi-view geometry. This is a grand challenge, especially when dealing with partial visibility due to occlusions and missing detections. Here we provide a complete characterization for the case of complete multi-view visibility.

Informally, a minimal problem is a 3D reconstruction problem that recovers camera poses and world coordinates from given images such that random input instances have a finite positive number of solutions. In particular, random measurement errors do not change the number of solutions.

Contribution: We give a complete classification of minimal problems for generic arrangements of points and lines, including their incidences, completely observed by any number of calibrated perspective cameras. We consider calibrated scenarios, as it avoids many degeneracies [40].

We show that there are exactly 30 minimal point-line problems (up to an arbitrary number of lines in the case of two views) when considering complete visibility (Table I). In particular, there is no such minimal problem for seven or more cameras. For 6, 5, 4 and 3 cameras, there are 1, 3, 6 and 17 minimal problems, respectively. For two views, there are three combinatorial constellations of five points which yield minimal problems. We observe that each minimal point-line problem has at most five points and at most six lines (except for arbitrarily many lines in the case of two views). Problems 5000_2 [1], 3200_3 [41], [42], $3010_0, 1040_0$ [43] have been known before, all other 26 minimal problems in Table I, as far as we know, are new.

For each minimal problem, we compute its *algebraic degree* which is its number of solutions over the complex numbers for generic images. This degree measures the intrinsic difficulty of a minimal problem. We observe how this degree generally grows with the number of cameras, but we also found several minimal problems with small degrees (32, 40 and 64), which might be practical in image matching and 3D reconstruction [32].

We consider generic minimal problems, i.e., the problems that have a finite number of complex solutions and are generic in the sense that random noise in image measurements does not change the number of solutions. For instance, the classical problem of five points in two views [1] is minimal, and one can add arbitrarily many lines to the arrangement in 3-space; as long as it contains five points in sufficiently generic position, it is still minimal. On the other hand, the problem of four points in three views [44], [45], [46], [47] is overconstrained when all measurements and equations are used. It becomes inconsistent for noisy image measurements. Thus, it is not a minimal problem for us.

Manuscript received 6 June 2022; revised 18 September 2023; accepted 3 October 2023. Date of publication 16 October 2023; date of current version 5 December 2023. The work of Timothy Duff and Anton Leykin was supported in part by the NSF under Grants DMS #1719968 and #2001267. The work of Timothy Duff was also supported by the NSF Mathematical Sciences Postdoctoral Reserarch Fellowship under Grant DMS-2103310. The work of Kathlén Kohn was supported by the Knut and Alice Wallenberg Foundation within their WASP (Wallenberg AI, Autonomous Systems and Software Program) AI/Math initiative. The work of Tomas Pajdla was supported in part by EU Regional Development Fund IMPACT under Grants CZ.02.1.01/0.0/0.0/15 003/0000468 and EU H2020, and in part by SPRING under Grant 871245 projects. This work was supported by the ICERM (NSF DMS-1439786 and the Simons Foundation under Grant 507536) for the hospitality (09/2018 – 02/2019), where most ideas for this project were developed. Recommended for acceptance by Y. Furukawa. *(Corresponding author: Timothy Duff.)*

Timothy Duff is with the Department of Mathematics, University of Washington, Seattle, WA 98195 USA (e-mail: timduff@uw.edu).

Kathlén Kohn is with the KTH Royal Institute of Technology, 11428 Stockholm, Sweden (e-mail: kathlen@kth.se).

Anton Leykin is with the School of Mathematics, Georgia Tech, Atlanta, GA 30332 USA (e-mail: anton.leykin@gmail.com).

Tomas Pajdla is with Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University, Prague 16636, Czechia (e-mail: pajdla@cvut.cz).

Digital Object Identifier 10.1109/TPAMI.2023.3324728

TABLE I
ALL BALANCED POINT-LINE PROBLEMS, MODULO ADDING ARBITRARILY MANY LINES TO THE PROBLEMS WITH 2 VIEWS

m views	6	6	6	5	5	5	4	4	4	4	4	4	4
$p^{\mathrm{f}}p^{\mathrm{d}}l^{\mathrm{f}}l^{\mathrm{a}}_{\alpha}$	1021_1	1013_3	1005_5	2011_1	2003_2	2003_3	1030_0	1022_2	1014_4	1006_6	3001_1	2110_0	2102_1
$(p,l,\mathcal{I})$													
Minimal Degree	Y $> 180k^*$	N	N	Y 11296^*	Y 26240^*	Y 11008^*	Y 3040^*	Y 4512^*	N	N	Y 1728^*	Y 32^*	Y 544^*

m views	4	3	3	3	3	3	3	3	3	3	3	3	3
$p^{\mathrm{f}}p^{\mathrm{d}}l^{\mathrm{f}}l^{\mathrm{a}}_{\alpha}$	2102_2	1040_0	1032_2	1024_4	1016_6	1008_8	2021_1	2013_2	2013_3	2005_3	2005_4	2005_5	3010_0
$(p,l,\mathcal{I})$													
Minimal Degree	Y 544^*	Y 360	Y 552	Y 480	N	N	Y 264	Y 432	Y 328	Y 480	Y 240	Y 64	Y 216

m views	3	3	3	3	3	3	3	3	2	2	2	2	2
$p^{\mathrm{f}}p^{\mathrm{d}}l^{\mathrm{f}}l^{\mathrm{a}}_{\alpha}$	3002_1	3002_2	2111_1	2103_1	2103_2	2103_3	3100_0	2201_1	5000_2	4100_3	3200_3	3200_4	2300_5
$(p,l,\mathcal{I})$													
Minimal Degree	Y 312	Y 224	Y 40	Y 144	Y 144	Y 144	Y 64	N	Y 20	Y 16	Y 12	N	N

Some problems are not uniquely identified by their vector $(p^f; p^d; l^f; l^a)$. To make the identification unique, we extend the vector by a subscript α, which is the maximum number of lines adjacent to the same point in the case of at least three views or the maximum number of points on a common line in the case of two views. Degrees marked with * have been computed with numerical methods, the others with symbolic algorithms; see Section 7. Problem 3200_3 has all five points in a single 3D plane: it corresponds to the calibrated homography relative pose computation [41], [42]; see Section 8.1

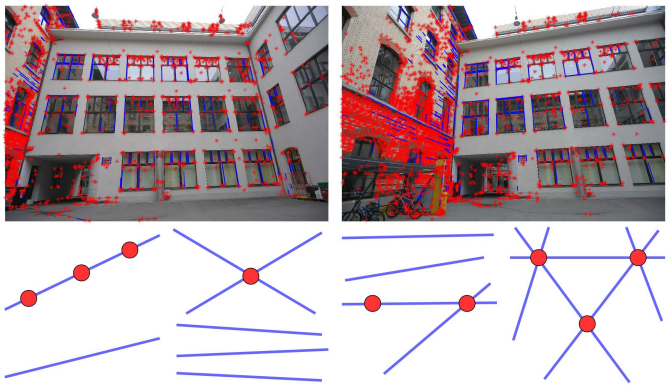


Fig. 1. (1-st row) Points (red) and lines (blue) get detected independently as well as in arrangements with points incident to lines [25], [39]. (2-nd row) Examples of some interesting arrangements of points and lines providing new minimal problems. See Table I for the complete classification of minimal problems for points, lines and their arrangements in multiple images with complete multi-view visibility.

We assume complete visibility, i.e.all points and lines are observed in all images, and all observed information is used to formulate minimal problems. Complete point-line incidence correspondences arise when, e.g., SIFT point features [48] are considered together with their orientation, lines are constructed from matched affine frames [49], or obtained as simultaneous point and line detections [25], Fig. 1.[1] On the other hand, we do not cover cases that require partial visibility, e.g., 3 PPP + 1 PPL in [43]. Full account for partial visibility is a much harder task and will be addressed in the future.

We explicitly model point-line incidences. Several lines may be incident to a single point (n-quiver) and several points may be dependent by lying on a single line. We assume that such relations are not broken by image noise, since they are constructed by the feature detection process.

Our problem formulation uses direct geometrical determinantal constraints as in [50], [51], not multi-view tensors, since it works for any number of cameras and can model point-line incidences. On the other hand, our formulation is not the most economical for the minimal problem with five independent points in two views (5000_2 in Table I). This problem has degree 20 in our formulation while the degree is only 10 when reformulated [52] as a problem of finding the essential matrix.[2]

The first version of this paper appeared in [54]. Here we provide additional details: (i) we discuss planar configurations of points in space, (ii) give full implementation details on how to compute the algebraic degree and check for minimality of a problem, and (iii) show concrete examples that allow us to reproduce our results.

Structure of the Paper: The paper is organized as follows. We review previous work in Section II. Section III defining the main concepts is followed by the problem specification in Section IV. All candidates for minimal problems satisfying balanced counts

[1] Real images from [39] were provided S. Ramalingam [25].

[2] Proving that the minimal problems with three or more views cannot be reformulated in a way that decreases the degrees that we report, or finding reformulations, may require more advanced algebraic techniques and presents a challenge for the future research. Preliminary results in this direction have already been reported in [53].

of degrees of freedom are identified in Section V. Section VI presents our parameterization of the problems for computational purposes. Procedures for checking minimality and computing degrees using symbolic and numerical methods from algebraic geometry are outlined in Section VII. Full implementation details and a discussion on planar point configurations are given in Section VIII.

II. Related Work

Here we review the most relevant work for point-line incidences and minimal problems. See [19], [20], [43] for references on minimal problems in general. Using correspondences of non-incident points and lines in three uncalibrated views was considered in works on the trifocal tensor [55]. The early work on point-line incidences [50] introduced n-quivers, i.e.points incident with n lines in uncalibrated views, and studied minimal problems arising from three 1-quivers in three affine views and three 3-quivers in three perspective views, as well as the overconstrained problem of four 2-quivers in three views.

Uncalibrated multi-view constraints for points, lines and their incidences appeared in [56]. In [51], non-incident points and lines in uncalibrated views were studied and four points and three lines in three views, two points and six lines in three views, and nine lines in three views cases were presented. The solver for the latter case has recently appeared in [57]. Absolute pose of cameras with unknown focal length from 2D-3D quiver correspondences has been solved in [11] for two points and 1-quiver, for one 1-quiver and one 2-quiver, and for four lines. See the discussion in [11] for more details on other absolute pose methods based on points and lines.

In [58], [59], an important case, when lines incident to points arise from tangent lines to curves, is presented. It motivates the case with three points and tangent lines at two points (case 3002_1 in Table I). Work [25] presents several minimal problems for generalized camera absolute pose computation from 2D-3D correspondences of non-incident points and lines with focus on cases when a closed-form solution could be found. In [6], [15], parallelism and perpendicularity of lines in space were exploited to find calibrated relative pose from lines and points. Recent work [60] investigates calibrated relative camera pose problems from two views with 2-quivers (called ray-point-ray, RPR, features) with known angles between the 3D lines generating the quivers. Minimal problems for finding the relative pose from three such correspondences for the generic as well as several more specific cases is derived. Our closest generalization of this result is that one can obtain calibrated relative pose of three cameras from one 2-quiver and two independent points in three views without knowing angles in 3D. Recently, minimal problems were constructed for local multi-features including lines incident to points as well as more complex features [22], [23], [24], [61], [62]. They build on SIFT directions [48] or more elaborate local affine features [49] to reduce the number of samples needed in RANSAC [63] to verify tentative matches.

The Most Relevant Previous Work: Recent theoretical results [21], [64], [65], [66], [67], [68] made steps towards characterizing some of the classes of minimal problems. The most relevant work [43] provided a classification for three calibrated views that can be formulated using linear constraints on the trifocal tensor [40].

In [43], 66 minimal problems for three calibrated views were presented and their algebraic degrees computed. The lowest degree 160 has been observed for one PPP and four PPL constraints while the highest degree 4912 has been observed for 11 PLL constraints. Classification [43] missed many minimal problems for complete visibility since they cannot be formulated using linear constraints on the trifocal tensor. Out of 66 problems in [43] the ones that can be modeled with complete visibility are (1 PPP + 4 LLL) and (3 PPP + 1 LLL). These two minimal problems appear as 1040_0 and 3010_0 in Table I. The other 15 minimal problems in three views that we discovered do not appear in [43] since the point-line incidences were not considered.

Work [69] extended the classification of [43] to many more calibrated trifocal minimal problems. It shows that infinitely many minimal problems under partial visibility (and when each line is incident to at most one point) can be grouped into 140616 equivalence classes by removing superfluous features and relabeling the cameras. The work used techniques and approaches developed in the conference version of this paper [54].

III. Notation and Concepts

We use nomenclature from [40] for basic concepts in geometry of computer vision. See [70] for the fundamentals of algebraic geometry, including Gröbner bases.

$\mathrm{SO}(3)$ stands for the special orthogonal group, i.e.rotations, defined algebraically as 3×3 matrices R such that $RR^\top = I$ and $\det(R) = 1$. We note that the dimension of $\mathrm{SO}(3)$ is three.

For u in $\mathbb{R}^3$, the skew-symmetric matrix $[u]_\times$ in $\mathbb{R}^{3\times 3}$ represents the cross product with u in $\mathbb{R}^3$, i.e.$[u]_\times v = u \times v$ for all v in $\mathbb{R}^3$.

Points in space are in the projective space $\mathbb{P}^3$, whereas image points are in $\mathbb{P}^2$. So points are represented by homogeneous coordinates.

We sometimes refer to the concept of algebraic varieties. A *projective variety* is the common zero set of a system of homogeneous polynomial equations. If the polynomials are defined in $N+1$ homogeneous variables, the projective variety lives in $\mathbb{P}^N$. Similarly, an *affine variety* is the common zero set of a system of polynomial equations which are not necessarily homogeneous. If the polynomials are defined in N variables over the ground field $\mathbb{F}$, the affine variety lives in $\mathbb{F}^N$. For any subset S of either $\mathbb{P}^N$ or $\mathbb{F}^N$, the *Zariski closure* $\overline{S}$ of S is the smallest projective, resp. affine, variety containing S.

The Grassmannians $\mathbb{G}_{1,3}$ and $\mathbb{G}_{1,2}$, which are the sets of all lines in $\mathbb{P}^3$ and $\mathbb{P}^2$, respectively, are examples of smooth projective varieties. The Grassmannian $\mathbb{G}_{1,2}$ of lines in $\mathbb{P}^2$ is isomorphic to $\mathbb{P}^2$ as every line in the projective plane is uniquely defined by a single linear equation with three homogeneous coordinates as coefficients. The Grassmannian $\mathbb{G}_{1,3}$ of lines in $\mathbb{P}^3$ can be seen as a projective variety via its embedding into $\mathbb{P}^5$ defined by its *Plücker coordinates*.

A line in $\mathbb{P}^3$ is defined by two linear equations. We write the homogeneous coefficients of these two equations as the two rows of a 2×4 matrix $\begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \end{bmatrix}$. The six maximal minors

of this matrix form the Plücker coordinates of the line in $\mathbb{P}^3$: $p_{ij} = a_i b_j - a_j b_i$. Every line in $\mathbb{P}^3$ is uniquely defined by its six Plücker coordinates. Moreover, the Plücker coordinates of a line in $\mathbb{P}^3$ satisfy the equation $p_{01}p_{23} - p_{02}p_{13} + p_{03}p_{12} = 0$. In addition, every projective tuple $(p_{01} : p_{02} : p_{03} : p_{12} : p_{13} : p_{23}) \in \mathbb{P}^5$ satisfying the equation $p_{01}p_{23} - p_{02}p_{13} + p_{03}p_{12} = 0$ is the Plücker coordinates of a line in $\mathbb{P}^3$.

This shows that the Grassmannian $\mathbb{G}_{1,3}$ is the zero set of the polynomial equation $p_{01}p_{23} - p_{02}p_{13} + p_{03}p_{12} = 0$ in $\mathbb{P}^5$. Furthermore, the Plücker coordinates of a line in $\mathbb{P}^3$ serve as homogeneous coordinates defining the line. So we can represent both points and lines in $\mathbb{P}^2$ as well as $\mathbb{P}^3$ by homogeneous coordinates.

A variety X is said to be *irreducible* if it is not the union of two proper subvarieties, i.e.if it cannot be written as $X = X_1 \cup X_2$ where X_1 and X_2 are non-empty subvarieties of X which are not equal to X. For instance, Grassmannians are irreducible.

Throughout our article, we consider an arbitrary ground field $\mathbb{F}$ unless explicitly specified. For different purposes we use different fields. For instance, the minimal problems we study are clearly defined over the field $\mathbb{R}$ of real numbers. When setting up systems of polynomial equations, coefficients originate from the field $\mathbb{Q}$ of rational numbers. Solutions of the equations are in the field $\mathbb{C}$ of complex numbers. We carry out symbolic computations in a finite field $\mathbb{Z}_p$ for a prime p for the sake of exactness and computational efficiency. Numerical algorithms use floating point to approximate complex numbers.

IV. Problem Specification

Our main result applies to problems in which points, lines, and point-line incidences are observed. We first introduce a *point-line problem* as a tuple $(p, l, \mathcal{I}, m)$ specifying that p points and l lines in space, which are incident according to a given incidence relation $\mathcal{I} \subset \{1, \dots, p\} \times \{1, \dots, l\}$ (i.e.$(i, j) \in \mathcal{I}$ means that the ith point is on the jth line) are projected to m views. So a point-line problem captures the numbers of points, lines and views as well as the incidences between points and lines. We will model intersecting lines by requiring that each intersection point of two lines has to be one of the p points in the point-line problem. Throughout this article we will only consider incidence relations which can be realized by a point-line arrangement in $\mathbb{P}^3$. In particular, two distinct lines cannot be incident to the same two distinct points. In addition, we will always assume that the incidence relation $\mathcal{I}$ is complete in the sense that every incidence which is automatically implied by the incidences in $\mathcal{I}$ must also be contained in $\mathcal{I}$. An *instance* of a point-line problem is specified by the following data:

1) A point-line arrangement in space consisting of p points $X_1, \dots, X_p$ and l lines $L_1, \dots, L_l$ in $\mathbb{P}^3$ which are incident exactly as specified by $\mathcal{I} \subset \{1, \dots, p\} \times \{1, \dots, l\}$. Hence, the point X_i is on the line L_j if and only if $(i, j) \in \mathcal{I}$. We write
$$\mathcal{X}_{p,l,\mathcal{I}} = \left\{(X, L) \in \left(\mathbb{P}^3\right)^p \times \left(\mathbb{G}_{1,3}\right)^l \mid \forall (i, j) \in \mathcal{I} : X_i \in L_j\right\}$$
for the associated *variety of point-line arrangements*. Note that this variety also contains degenerate arrangements, where not all points and lines have to be pairwise distinct or where there are more incidences between points and lines than those specified by $\mathcal{I}$.
2) A list of m calibrated cameras which are represented by matrices
$$P_1 = [R_1 \mid t_1], \dots, P_m = [R_m \mid t_m]$$
with $R_1, \dots, R_m \in \mathrm{SO}(3)$ and $t_1, \dots, t_m \in \mathbb{F}^3$.
3) The *joint image* consisting of the projections $x_{v,1}, \dots, x_{v,p} \in \mathbb{P}^2$ of the points $X_1, \dots, X_p$ and the projections $\ell_{v,1}, \dots, \ell_{v,l} \in \mathbb{G}_{1,2}$ of the lines $L_1, \dots, L_l$ by the cameras $P_1, \dots, P_m$ to the $v = 1, \dots, m$ views. We write
$$\mathcal{Y}_{p,l,\mathcal{I},m} = \Big\{(x, \ell) \in \left(\mathbb{P}^2\right)^w \times \left(\mathbb{G}_{1,2}\right)^{m\,l} \;\Big|\; \forall v = 1, \dots, m \;\; \forall (i, j) \in \mathcal{I} : x_{v,i} \in \ell_{v,j}\Big\}$$
for the *image variety* which consists of all m-tuples of two-dimensional point-line arrangements which satisfy the incidences specified by $\mathcal{I}$. This is a generalization of joint image previously introduced in [71], [72].

Given a joint image, we want to recover an arrangement in space and cameras yielding the given joint image. We refer to a pair of such an arrangement and such a list of m cameras as a *solution* of the point-line problem for the given joint image. We note that an m-tuple in $\mathcal{Y}_{p,l,\mathcal{I},m}$ does not necessarily admit a solution, i.e.a priori it does not have to be a joint image of a common point-line arrangement in 3-space.

To fix the arbitrary space coordinate system [40], we set $P_1 = [I \mid 0]$ and the first coordinate of t_2 to 1. Hence, our *camera configurations* are parameterized by
$$\mathcal{C}_m = \left\{(P_1, \dots, P_m) \in \left(\mathbb{F}^{3\times 4}\right)^m \;\middle|\; P_i = [R_i \mid t_i], \; R_i \in \mathrm{SO}(3), t_i \in \mathbb{F}^3, \; R_1 = I, t_1 = 0, t_{2,1} = 1\right\}.$$

We will always assume that the camera positions in an instance of a point-line problem are sufficiently generic such that the following three natural conditions are satisfied for each camera: First, two distinct lines or points in the given arrangement in 3-space are viewed as distinct lines or points. Second, a point and a line in the space arrangement, which are not incident in 3-space, are viewed as non-incident. Third, three non-colinear points in the space arrangement are viewed as non-colinear points.

We say that a point-line problem is *minimal* if a generic image tuple in $\mathcal{Y}_{p,l,\mathcal{I},m}$ has a nonzero finite number of solutions. We may phrase this definition formally:

Definition 1: Let $\Phi_{p,l,\mathcal{I},m} : \mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m \dashrightarrow \mathcal{Y}_{p,l,\mathcal{I},m}$ denote the *joint camera map*, which sends a point-line arrangement in space and m cameras to the resulting joint image. We say that the point-line problem $(p, l, \mathcal{I}, m)$ is *minimal* if

- $\Phi_{p,l,\mathcal{I},m}$ is a *dominant map*,[3] i.e. a generic element (x, ℓ) in $\mathcal{Y}_{p,l,\mathcal{I},m}$ has a solution, so $\Phi^{-1}_{p,l,\mathcal{I},m}(x, \ell) \neq \emptyset$, and

[3]Dominant maps are analogs of surjective maps in birational geometry.

• the preimage $\Phi^{-1}_{p,l,\mathcal{I},m}(x,\ell)$ of a generic element (x,ℓ) in $\mathcal{Y}_{p,l,\mathcal{I},m}$ is finite.

Remark 1: For a given a minimal problem $(p,l,\mathcal{I},m)$, the joint camera map $\Phi_{p,l,\mathcal{I},m}$ maps $\mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m$ onto a constructible subset of $\mathcal{Y}_{p,l,\mathcal{I},m}$ of the same dimension. Given a solution for a generic joint image (x,ℓ) when $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, there exists a ball around (x,ℓ), say $B_\epsilon(x,\ell)$, and for each solution $(X,C) \in \Phi^{-1}_{p,l,\mathcal{I},m}(x,\ell)$ a ball $B_\delta(X,C)$ such that

$$\Phi_{p,l,\mathcal{I},m}\left(B_\delta(X,C)\right) \subset B_\epsilon(x,\ell).$$

In this sense, we may deduce that solutions to minimal problems are stable under perturbation of the data.

The joint camera map $\Phi_{p,l,\mathcal{I},m}$ reflects that we want to recover world points and lines as well as camera poses from a given joint image. In the case of complete visibility, this is equivalent to only recovering camera poses. We formalize this observation in Lemma 2 and Corollary 2.

Over the complex numbers, the cardinality of the preimage $\Phi^{-1}_{p,l,\mathcal{I},m}(x,\ell)$ is the same for every *generic* joint image (x,ℓ) of a minimal point-line problem $(p,l,\mathcal{I},m)$. We refer to this cardinality as the *degree* of the minimal problem. Our goal is to list *all* minimal point-line problems and to compute their degrees. For this, we pursue the following strategy:

Step 1: A classical statement from algebraic geometry states for a dominant map $\varphi : X \dashrightarrow Y$ from a variety X to another variety Y that the preimage $\varphi^{-1}(y)$ of a generic point y in Y has dimension $\dim(X) - \dim(Y)$. When φ is a linear map between linear spaces, this is simply the rank-nullity theorem of linear algebra. As the generic preimage of the joint camera map $\Phi_{p,l,\mathcal{I},m}$ associated to a minimal point-line problem $(p,l,\mathcal{I},m)$ is zero-dimensional, we see that every minimal point-line problem must satisfy the equality $\dim(\mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m) = \dim(\mathcal{Y}_{p,l,\mathcal{I},m})$. This motivates the following definition.

Definition 2: We say that a point-line problem $(p,l,\mathcal{I},m)$ is *balanced* if $\dim(\mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m) = \dim(\mathcal{Y}_{p,l,\mathcal{I},m})$.

As we have now established that all minimal point-line problems are balanced, we classify all balanced point-line problems in Section V. We will see that there are only finitely many such problems, explicitly given in Table I, up to arbitrarily many lines in the case of two views; see Remark 2.

Step 2: The classical statement from algebraic geometry mentioned above further implies that a balanced point-line problem $(p,l,\mathcal{I},m)$ is minimal if and only if its joint camera map $\Phi_{p,l,\mathcal{I},m}$ is dominant. Hence, to determine the exhaustive list of all minimal point-line problems, we only have to check for each balanced point-line problem in Table I if its joint camera map is dominant. We perform this check computationally, as described in Section VII.

Step 3: Finally, we use symbolic and numerical computations to calculate the degrees of the minimal point-line problems. We describe these computations in Section VII.

V. Balanced Point-Line Problems

To understand balanced point-line problems we need to derive formulas for the dimensions of the varieties $\mathcal{X}_{p,l,\mathcal{I}}$, $\mathcal{C}_m$ and $\mathcal{Y}_{p,l,\mathcal{I},m}$. As $\mathrm{SO}(3)$ is three-dimensional, and we set the first camera to $[I|0]$ and one parameter in the second camera to 1, the parameter space of camera configurations for $m \geq 2$ has dimension $\dim(\mathcal{C}_m) = 6m - 7$.

Let us now consider a generic point-line arrangement in $\mathcal{X}_{p,l,\mathcal{I}}$. Some of its points may be dependent on other points, in the sense that such a dependent point lies on a line spanned by two other points. In any arrangement of points in 3-space, each minimal set of independent points has the same cardinality. For our arrangement of p points we denote this cardinality by p^{f} (the upper index f stands for *free*). We write $p^{\mathrm{d}} = p - p^{\mathrm{f}}$ for the number of dependent points. Each free point is defined by three parameters. A dependent point X is only defined by *one* further parameter after the two points, which span the line containing X, are defined. In total, the p points in our arrangement are defined by $3\,p^{\mathrm{f}} + p^{\mathrm{d}}$ parameters. Each of the l lines in our arrangement is either incident to zero, one or at least two points. We refer to lines which are incident to no points as *free lines*. We denote the number of free lines by l^{f}. As the Grassmannian $\mathbb{G}_{1,3}$ of lines is four-dimensional, each free line is defined by four parameters. A line which is incident to a fixed point is defined by only two parameters. We denote the number of lines which are incident to exactly one point by l^{a} (the upper index a stands for *adjacent*). Finally, each of the remaining $l - l^{\mathrm{f}} - l^{\mathrm{a}}$ lines is incident to at least two points and thus already uniquely determined by the two points. Hence, we have derived

$$\dim(\mathcal{X}_{p,l,\mathcal{I}}) = 3\,p^{\mathrm{f}} + p^{\mathrm{d}} + 4\,l^{\mathrm{f}} + 2\,l^{\mathrm{a}}. \tag{1}$$

In particular, we see that we might as well assume that there is no line passing through two or more points, as such lines do not contribute to our parameter count.

We derive the dimension of the image variety $\mathcal{Y}_{p,l,\mathcal{I},m}$ similarly. Since we assume all camera positions to be sufficiently generic, each camera views exactly p^{f} independent points, p^{d} dependent points, l^{f} free lines and l^{a} lines which are incident to exactly one of the points. Each independent point is defined by two parameters, whereas each dependent point is defined by a single parameter. A free line is defined by two parameters. A line which is incident to a fixed point is defined by a single parameter. All in all, we have that

$$\dim(\mathcal{Y}_{p,l,\mathcal{I},m}) = m\,(2\,p^{\mathrm{f}} + p^{\mathrm{d}} + 2\,l^{\mathrm{f}} + l^{\mathrm{a}}). \tag{2}$$

Note that there is no balanced point-line problem for a single camera. For $m > 1$ cameras, combining $\dim(\mathcal{C}_m) = 6m - 7$ with (1) and (2) yields that a point-line problem is balanced if and only if

$$3\,p^{\mathrm{f}} + p^{\mathrm{d}} + 4\,l^{\mathrm{f}} + 2\,l^{\mathrm{a}} + 6\,m - 7 = m\left(2\,p^{\mathrm{f}} + p^{\mathrm{d}} + 2\,l^{\mathrm{f}} + l^{\mathrm{a}}\right).$$

This is equivalent to

$$6m-7=(2m-3)p^{\mathrm{f}}+(m-1)p^{\mathrm{d}}+2(m-2)l^{\mathrm{f}}+(m-2)l^{\mathrm{a}}. \tag{3}$$

Lemma 1: Every balanced point-line problem with at least five points has exactly two cameras.

Proof: Suppose $(p,l,\mathcal{I},m)$ is a balanced point-line problem with $m > 1$ cameras and at least five points, i.e. $p^{\mathrm{f}} + p^{\mathrm{d}} \geq 5$. In

this case, the equality (3) implies

$$6m-7 \geq (2m-3)p^{\mathrm{f}}+(m-1)(5-p^{\mathrm{f}})=(p^{\mathrm{f}}+5)m-(2p^{\mathrm{f}}+5),$$

which is equivalent to

$$2(p^{\mathrm{f}}-1) \geq (p^{\mathrm{f}}-1)m. \tag{4}$$

Among the five or more points at least two have to be (by definition) independent, i.e.$p^{\mathrm{f}} > 1$. So (4) yields $m \leq 2$. □

Theorem 1: There is no balanced point-line problem with seven or more cameras.

Proof: Let $(p, l, \mathcal{I}, m)$ be a balanced point-line problem with $m \geq 7$ cameras. By equality (3), we have

$$5 \equiv p^{\mathrm{f}} + p^{\mathrm{d}} \mod (m-2). \tag{5}$$

This implies $p^{\mathrm{f}} + p^{\mathrm{d}} \geq 5$ if $m \geq 8$, which contradicts Lemma 1, and thus we have only one remaining case to check: $m = 7$. From (5) and Lemma 1, we have $p^{\mathrm{f}} + p^{\mathrm{d}} = 0$ in the case of seven cameras. It means that there are no points, and thus there cannot be lines which are incident to points. So we have $p^{\mathrm{f}} = 0$, $p^{\mathrm{d}} = 0$, $l^{\mathrm{a}} = 0$, and (3) reduces to $35 = 10l^{\mathrm{f}}$, which is clearly not possible. So there are no balanced point-line problems with seven or more cameras. □

Theorem 2: There are 34 balanced point-line problems with 3, 4, 5 or 6 cameras. They are all listed in Table I.

Proof: We consider the different cases for $3 \leq m \leq 6$ and reason by cases.

- $m = 6$: Due to (5) and Lemma 1, every balanced point-line problem with six cameras must have exactly one point. So we have $p^{\mathrm{f}} = 1$, $p^{\mathrm{d}} = 0$, and (3) reduces to $5 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us three possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(2,1),(1,3),(0,5)\}$ (see first row of Table I).
- $m = 5$: Due to (5) and Lemma 1, every balanced point-line problem with five cameras must have exactly two points. So we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 0$, and (3) reduces to $3 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us two possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1,1),(0,3)\}$, which yield three point-line problems (see the first row of Table I).
- $m = 4$: Due to (5) and Lemma 1, every balanced point-line problem with four cameras must have either one point or three points. Let us first consider the case of a single point. Here we have $p^{\mathrm{f}} = 1$, $p^{\mathrm{d}} = 0$, and (3) reduces to $6 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us four possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(3,0),(2,2),(1,4),(0,6)\}$ (see first row of Table I). Second, we consider balanced point-line problems with four cameras and three points. If all three points are independent, (3) reduces to $1 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$, which has a single solution: $(l^{\mathrm{f}}, l^{\mathrm{a}}) = (0,1)$. If not all three points are independent, we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 1$, and (3) reduces to $2 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us two possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1,0),(0,2)\}$, which yield three point-line problems (see the first two rows of Table I for all four point-line problems with four cameras and three points).
- $m = 3$: We first observe that each balanced point-line problem with three cameras must have at least one point. Otherwise we would have $p^{\mathrm{f}} = 0, p^{\mathrm{d}} = 0$ and $l^{\mathrm{a}} = 0$, so (3) would reduce to $11 = 2l^{\mathrm{f}}$, which is impossible. Let us first consider the case of a single point. Here we have $p^{\mathrm{f}} = 1$, $p^{\mathrm{d}} = 0$, and (3) reduces to $8 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us five possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(4,0),(3,2),(2,4),(1,6),(0,8)\}$ (see second row of Table I). Second, in the case of two points, we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 0$, and (3) reduces to $5 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. This gives us three possibilities: $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(2,1),(1,3),(0,5)\}$, which yield six point-line problems (see second row of Table I). Third, we consider the case of three points. If all three points are independent, (3) reduces to $2 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. The two solutions $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1,0),(0,2)\}$ yield three point line problems (see last two rows of Table I). If not all three points are independent, we have $p^{\mathrm{f}} = 2$, $p^{\mathrm{d}} = 1$, and (3) reduces to $3 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$. The two solutions $(l^{\mathrm{f}}, l^{\mathrm{a}}) \in \{(1,1),(0,3)\}$ yield four point-line problems (see last row of Table I). Finally, we consider balanced point-line problems with three cameras and four points. We see from (3) that not all four points can be independent. Hence, we either have $p^{\mathrm{f}} = 3$ and $p^{\mathrm{d}} = 1$ such that (3) reduces to $0 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$, which has a single solution $(l^{\mathrm{f}}, l^{\mathrm{a}}) = (0,0)$, or we have $p^{\mathrm{f}} = 2$ and $p^{\mathrm{d}} = 2$ such that (3) reduces to $1 = 2l^{\mathrm{f}} + l^{\mathrm{a}}$, which also has a single solution $(l^{\mathrm{f}}, l^{\mathrm{a}}) = (0,1)$ (see the last row of Table I) □

Remark 2: For the case of two cameras, we see from (3) that the number of free and incident lines do not contribute to the parameter count for balanced point-line problems. In fact, (3) reduces for $m = 2$ to $5 = p^{\mathrm{f}} + p^{\mathrm{d}}$. Hence, we have the classical minimal problem of recovering five points from two camera images. More precisely, a point-line problem with two cameras is balanced if and only if it has five points. Therefore, it is irrelevant how many lines are contained in the arrangement or how many points are independent. There are 5 combinatorial possibilities to distribute dependent and independent points (see the last row of Table I).

Corollary 1: There are 39 balanced point-line problems, modulo any number of lines in the case of two views. They are listed in Table I.

VI. Eliminating World Points and Lines

In order to do computations, it is customary to describe problems with implicit equations that do not depend on the world variables. Before we describe such equations, let us phrase the elimination of the world variables geometrically.

We consider the Zariski closure of the graph of the joint camera map:

$$\mathrm{Inc} = \overline{\{(X,C,Y) \in \mathcal{X}_{p,l,\mathcal{I}} \times \mathcal{C}_m \times \mathcal{Y}_{p,l,\mathcal{I},m} \mid Y = \Phi_{p,l,\mathcal{I},m}(X,C)\}}.$$

The joint camera map $\Phi_{p,l,\mathcal{I},m}$ is dominant if and only if the projection $\pi_{\mathcal{Y}} : \mathrm{Inc} \to \mathcal{Y}_{p,l,\mathcal{I},m}$ onto the last factor is dominant (since this is the projection from the graph of $\Phi_{p,l,\mathcal{I},m}$ on its codomain). Moreover, the cardinality of the preimage of a generic point $Y \in \mathcal{Y}_{p,l,\mathcal{I},m}$ under both maps $\Phi_{p,l,\mathcal{I},m}$ and $\pi_{\mathcal{Y}}$ is the same.

To make computations simpler, we want to derive the same statement for the following restricted incidence variety, which

does not include the 3D structure $\mathcal{X}_{p,l,\mathcal{I}}$:

$$\mathrm{Inc}' = \overline{\{(C,Y) \in \mathcal{C}_m \times \mathcal{Y}_{p,l,\mathcal{I},m} \mid \overline{\exists X \in \mathcal{X}_{p,l,\mathcal{I}} : Y = \Phi_{p,l,\mathcal{I},m}(X,C)\}}}.$$

We have the canonical projections on the right, where $\pi_{\mathcal{C},\mathcal{Y}}$ omits the first factor and $\pi'_{\mathcal{Y}}$ projects onto the last factor:

$$\begin{array}{ccc} \mathrm{Inc} & \xrightarrow{\pi_{\mathcal{Y}}} & \mathcal{Y}_{p,l,\mathcal{I},m} \\ {\scriptstyle \pi_{\mathcal{C},\mathcal{Y}}}\downarrow & \nearrow_{\pi'_{\mathcal{Y}}} & \\ \mathrm{Inc}' & & \end{array}$$

Lemma 2: If $m \geq 2$, a generic point $(C,Y) \in \mathrm{Inc}'$ has a single point in its preimage under $\pi_{\mathcal{C},\mathcal{Y}}$.

Proof: $Y = (x, \ell)$ consists of points $x = (x_{1,1}, \ldots, x_{m,p})$ and lines $\ell = (\ell_{1,1}, \ldots, \ell_{m,l})$ in the m views. Each point $x_{v,i} \in \mathbb{P}^2$ in a view v is pulled back via the vth camera to a line in 3-space. As $m \geq 2$, the m pull-back lines for generic[4] $x_{1,i}, \ldots, x_{m,i}$ intersect in a unique point in $\mathbb{P}^3$. Similarly, each line $\ell_{v,j}$ in a view v is pulled back via the vth camera to a plane in $\mathbb{P}^3$. As $m \geq 2$, the m generic pull-back planes for $\ell_{1,j}, \ldots, \ell_{m,j}$ intersect in a unique line in $\mathbb{P}^3$. □

Corollary 2: A balanced point-line problem $(p,l,\mathcal{I},m)$ is minimal if and only if the projection $\pi'_{\mathcal{Y}}$ is dominant. In that case, the degree of the minimal problem is the cardinality of the preimage ${\pi'_{\mathcal{Y}}}^{-1}(Y)$ of a generic joint image $Y \in \mathcal{Y}_{p,l,\mathcal{I},m}$ over the complex numbers.

Proof: As we have observed in *Step 2* in Section IV, a balanced point-line problem is minimal if and only if its joint camera map $\Phi_{p,l,\mathcal{I},m}$ is dominant. This happens if and only if $\pi_{\mathcal{Y}}$ is dominant. Due to Lemma 2, this is equivalent to that $\pi'_{\mathcal{Y}}$ is dominant. Similarly, for a generic $Y \in \mathcal{Y}_{p,l,\mathcal{I},m}$, the cardinalities of the preimages $\Phi^{-1}_{p,l,\mathcal{I},m}(Y)$, $\pi_{\mathcal{Y}}^{-1}(Y)$ and ${\pi'_{\mathcal{Y}}}^{-1}(Y)$ coincide due to Lemma 2. □

We note that it is possible to describe the variety Inc' as a component of the variety cut out by the equations that we establish in the remainder of this section.

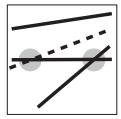
For any instance of a point-line problem, the solutions must satisfy certain equations defined in terms of joint images $(x,\ell) \in \mathcal{Y}_{p,l,\mathcal{I},m}$. Our scheme for generating such equations relies on an alternate representation of (x,ℓ) defined solely in terms of lines. The equations result from two types of constraints. The first type of constraint is a *line correspondence (LC)*: if $\ell_1, \ldots, \ell_m$ are images of the same world line, with respective homogeneous coordinates $\mathbf{l}_1, \ldots, \mathbf{l}_m \in \mathbb{F}^{3\times 1}$, then

$$\mathrm{rank}\begin{bmatrix} P_1^T\mathbf{l}_1 & P_2^T\mathbf{l}_2 & \ldots & P_m^T\mathbf{l}_m \end{bmatrix} \leq 2. \tag{6}$$

That is, the planes with homogeneous coordinates $P_i^T\mathbf{l}_i$ share a common line in $\mathbb{P}^3$. We distinguish two classes of lines in $\mathbb{P}^2$:

1) *Visible lines* define valid line correspondences. Besides $m\,l$ observed lines in the joint image, for generic x there is a unique visible line between any two observed points. Taken across all views, any pair of points thus provides a line correspondence which must be satisfied.
2) Two generic visible lines suffice to define a point. We may use an additional set of (non-corresponding) *ghost lines* to define any points which meet fewer than two visible lines. A generic ghost line contains exactly one observed point — it is simply a device for generating equations.[5]

Example (1)



Example (2)

Example (3)

Fig. 2. Encoding problems with visible and ghost lines.

Thus we obtain *common point (CP)* constraints: given visible and ghost lines $\mathbf{l}_{v,1}, \ldots \mathbf{l}_{v,k_i}$ which meet $x_{v,i}$, the projection of the ith point in the view $v \in 1, \ldots, m$, we must have

$$\mathrm{rank}\begin{bmatrix} P_1^T\mathbf{l}_{1,1} & \ldots & P_m^T\mathbf{l}_{m,k_i} \end{bmatrix} \leq 3, \quad i = 1, \ldots, p. \tag{7}$$

We may encode a point-line problem by specifying some number of visible lines, some number of ghost lines, and which of these lines are incident at each point. We illustrate this encoding with several examples appearing in Fig. 2:

Example 1:

1) Consider the point-line problem labeled "2013_2" in Table I. The lines explicitly drawn in the table together with a visible line between the two free points, as in Fig. 2, suffice to define the scene for generic data.
2) Consider now the problem labeled "2011_1" in Table I. The encoding given in Fig. 2 includes the given lines, a visible line between the given points, as well as a single ghost line needed to define one of the points.
3) Finally, consider the problem labeled "3200_3" in Table I. The extra visible lines appearing in Fig. 2 fix the positions of all points.

LC and CP constraints immediately translate into determinantal conditions: the 3×3 minors of the matrix in (6) must vanish for each visible line, and the 4×4 minors of the matrix in (7) must vanish for each point. Thus, we obtain explicit polynomials for each point-line problem once we fix some encoding and the *cameras parametrization*, i.e.a rational map $G : \mathbb{F}^{6m-7} \to \mathcal{C}_m$. In our computations, we define G via the Cayley parametrization for $\mathrm{SO}(3)$:

$$R([a,b,c]) = (I + [[a,b,c]]_\times)(I - [[a,b,c]]_\times)^{-1}. \tag{8}$$

[4]Genericity implies, for instance, that $x_{1,i}$ and $x_{2,i}$ are not epipoles.

[5]"Canonical" ghost lines, which are rows of $[x]_\times$, are often used to eliminate point x from equations by $[c]_\times\, x = 0$ [40], [56].

Algorithm 1: (Minimal).

Input: $(p, l, \mathcal{I}, m)$, a balanced point-line problem.
Output: "Y" if the problem is minimal; "N" otherwise.
1: $J(C, Y) \leftarrow \frac{\partial F(C,Y))}{\partial C}$
2: Take random $C_0 \in \mathcal{C}_m$ and random $X_0 \in \mathcal{X}_{p,l,\mathcal{I}}$.
3: $Y_0 \leftarrow \Phi_{p,l,\mathcal{I},m}(X_0, C_0)$
4: **return** "Y" if $\operatorname{rank} J(C_0, Y_0) = 6m - 7$, else "N."

Algorithm 2: (Degree).

Input: $(p, l, \mathcal{I}, m)$, a point-line minimal problem.
Output: The degree of this problem.
1: Take a random $Y_0 \in \mathcal{Y}_{p,l,\mathcal{I},m}$.
2: Compute the Gröbner basis B of the ideal generated by $F(C, Y_0) \subset \mathbb{F}[C]$.
3: **return** the number of monomials in variables C not divisible by the leading monomials of B.

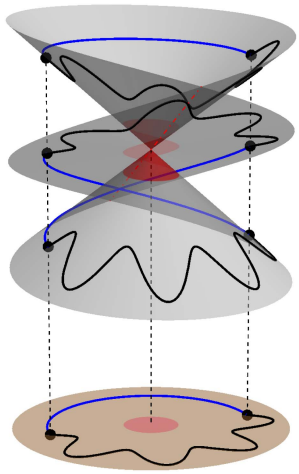


Fig. 3. Circulating along the path in the parameter space (downstairs) creates a path meeting all solutions (upstairs).

VII. Checking Minimality & Computing Degrees

Denote by $F = F_{p,l,\mathcal{I},m}$ the system of polynomials resulting from a given point-line problem $(p, l, \mathcal{I}, m)$ using our construction of the LC and CP constraints in Section VI with the cameras parameterization G plugged in. The variety of points satisfying $F(C, Y) = 0$ contains Inc$'$ as an irreducible component.

Remark 3: The variety of points satisfying $F(C, Y) = 0$ may also have spurious components corresponding to solutions (C, Y) where the ranks of the matrices (6) or (7) are smaller than desired. Such spurious solutions do not correspond to world lines in $\mathbb{G}_{1,3}$ and must be ruled out. These spurious components are naturally avoided by sampling a point on Inc$'$. For implicit symbolic calculations, the spurious solutions may be eliminated by including inequations enforcing nonvanishing of the minors of size one smaller.

The following algorithm checks minimality of a point-line problem locally; geometrically this amounts to passing to the tangent space of Inc$'$.

Proof of Correctness for Algorithm 1 In the terminology described in the beginning of Section VI, the algorithm checks if the conditions of the Inverse Function Theorem hold at a generic point on Inc$'$. If they do, the map $\pi'_{\mathcal{Y}}$ is dominant, since in a neighborhood of Y_0 the map has an inverse: i.e if Y is near Y_0 then there is C near C_0 satisfying $F(C, Y) = 0$. If these conditions do not hold generically, $\pi'_{\mathcal{Y}}$ is not dominant. By Corollary 2, the given point-line problem is minimal if and only if $\pi'_{\mathcal{Y}}$ is dominant. □

For the minimal problems with two and three views we use the following symbolic algorithm to compute their degrees, i.e., the cardinality of the preimage of a generic joint image $Y \in \mathcal{Y}_{p,l,\mathcal{I},m}$ under the projection $\pi'_{\mathcal{Y}}$ (by Corollary 2).

Proof of Correctness: Using Gröbner bases to solve a system of polynomial equations is a standard technique in computational nonlinear algebra. Since we are interested only in the solution count, *not* the solutions, we are able to carry out computations relatively quickly; see Remark 4. □

Remark 4: Algorithms 1 and 2 are valid over an arbitrary field $\mathbb{F}$. Our main problem is stated over $\mathbb{Q}$, the rational numbers, but since the algorithms rely heavily on symbolic techniques such as Gröbner bases we use the so-called *modular technique:* we perform computations over a finite field, namely $\mathbb{F} = \mathbb{Z}_p$ for $p < 2^{15}$. There is a slight chance that this approach fails for a particular exceptional "unlucky" prime p, but it is possible to compute the result using several primes and confirm it over $\mathbb{Q}$ via rational reconstruction.

Algorithms 1 and 2 were implemented and executed in the Macaulay2 [73] computer algebra system.[6] Due to limitations of Gröbner basis algorithms we were unable to compute the degrees of any of the problems with $m > 3$ with our implementation of Algorithm 2. On the other hand, the degrees of all minimal problems in Table I are within reach for the *monodromy method*, a technique based on numerical homotopy continuation. Specifically, we follow the *monodromy solver* framework outlined in [74] carrying out computation via a Macaulay2 package `MonodromySolver`[6]. Similar techniques have been successfully employed in a number of studies in applied algebraic geometry [75], [76], [77].

Imagine the projection $\pi'_{\mathcal{Y}} : \text{Inc}' \to \mathcal{Y}_{p,l,\mathcal{I},m}$ as the cover map from top to the bottom in Fig. 3. The seed solution (C_0, Y_0) produced as in Algorithm 1 is one of the solutions that project to Y_0 at the bottom. Since the Galois group of $\pi'_{\mathcal{Y}}$ acts transitively on the solutions, one can create enough random paths connecting Y_0 and an auxiliary point Y_1 so that walking on the liftings of the bottom paths, it is possible to visit all solutions that are above Y_0 and, hence, discover the degree.

VIII. Implementation Details

In this section, we describe the Macaulay2 [73] implementation of our computations through a series of guided examples. Each subsection provides commentary for a dedicated Macaulay2 file:

Besides these examples and setup code providing the core functionality, there are several scripts which automate the computations reported in the main text.

[6] Available at https://github.com/timduff35/PLMP

```
VIII-A "PLMP-3200_3.m2"
VIII-B "example-2111_1-unrolled.m2"
VIII-C "example-2111_1.m2"
VIII-D "example-2111_1-jacobian.m2"
VIII-E "example-2111_1-numerical.m2"
```

We begin by verifying that the constraints described in Section VI give the same number of solutions in (R, t) as classical approaches in the well-known planar calibrated homography problem; see Section VIII-A. In the remaining sections, we illustrate our computations on a running example. We give details for our symbolic degree computation and minimality check in Sections VIII-B and VIII-D. Section VIII-C demonstrates the automated setup script used for this problem. Finally, in Section VIII-E we give an overview of the numerical methods used to compute degrees for problems with more than three cameras.

A. *Example:* 3200_3 – *Planar Calibrated Homography*

Here we further investigate 3200_3 problem. More precisely, we show that the number 12 reported in our table agrees with the classical formulations.

Problem 3200_3 has five points on two lines with one point in common to both lines. Hence, all five points are in a single 3D plane. This corresponds to the relative calibrated camera pose from homography [41], [42]. It first computes a homography between the two images using four point correspondences. The fifths correspondence is determined by the four and automatically satisfies the homography. Having four correspondences, a single solution, $A \in \mathbb{R}^{3\times 3}$ viewed as in $\mathbb{P}^9$, is obtained and then decomposed into multiple possible rotations, translations, normals and a scales.

The decomposition problem can be formulated as follows [41]

$$A = d\,R + T\,N^T$$

with rotation R, translation T, normal N and scale d.

The decomposition of a homography matrix A, which was developed in [41], is based on the SVD. It generates 8, resp. 4, solution in a generic situations with $\|N\|^2 = 1$, resp. $N_3 = 1$, dehomogenization. The following Macaulay2 code shows that the degree over the complex numbers is 24, resp. 12, in a generic situations with $\|N\|^2 = 1$, resp. $N_3 = 1$.

```
Rng = QQ[r_(1,1)..r_(3,3),t_1..t_3,n_1..n_3,d]
R = transpose(genericMatrix(Rng,r_(1,1),3,3))
T = genericMatrix(Rng,t_1,3,1)
N = genericMatrix(Rng,n_1,3,1)
A = random(QQ^3,QQ^3)
I = ideal(A - d*R-T*transpose(N)) +
    ideal(transpose(R)*R-id_(QQ^3)) +
    ideal(det(R)-1) +
    ideal(transpose(N)*N-1)
dim I, degree I
o7 = (0, 24)

I = ideal(A - d*R-T*transpose(N)) +
    ideal(transpose(R)*R-id_(QQ^3)) +
    ideal(det(R)-1) +
    ideal(n_3-1)
dim I, degree I
o7 = (0, 12)
```

We used a linear constraint to dehomogenize when computing the degree of 3200_3 problem over the complex numbers and obtained its degree equal to 12, which is in agreement with the above homography formulation.

B. *Computing the Degree Symbolically "from Scratch"*

We work over the finite field $\mathbb{Z}_{10007}$:

$$\texttt{FF} = \texttt{ZZ}/10007$$

We define point data in $\mathbb{P}^2$ in three views over $\mathbb{Z}_{10007}$ consistent with the problem "2111_1"). This is represented as a list of three matrices in Macaulay2:

```
i2 : P = {matrix{{-2639,-4936,1789},
                 {2653,-591,-643},
                 {1,1,1}},
          matrix{{-3868,-1776,3174},
                 {3669,-4143,-1982},
                 {1,1,1}},
          matrix{{-1889,-1604,4629},
                  {-473,-4513,-4210},
                  {1,1,1}}};
```

The matrix is best viewed in an output buffer using the `netList` command:

```
i3 : netList P

     +---------------------+
o3 = || -2639 -4936 1789 | |
     || 2653  -591  -643 | |
     || 1     1     1    | |
     +---------------------+
     || -3868 -1776 3174  ||
     || 3669  -4143 -1982 ||
     || 1     1     1     ||
     +---------------------+
     || -1889 -1604 4629  ||
     || -473  -4513 -4210 ||
     || 1     1     1     ||
     +---------------------+
```

The first two columns of each matrix in `P` were randomly generated, while the last column is (projectively) a random $\mathbb{Z}_{10007}$-linear combination of the first two. This reflects the dependence of the third point on the first two in each view.

In general, we encode point-line problems by specifying the number of visible lines, the number of ghost lines, and, for each point, a list of labels of lines passing through it. For problem "2111_1" this is accomplished as follows:

```
D = (3,2,{{1,2},{1,3},{1,4}})
nLines = D#0 + D#1
lineIncidences = D#2
```

The list `L` contains random line matrices (homogeneous coordinates of lines as in rows) consistent with `P` and the encoding `D`:

```
i8 : netList L

     +---------------------+
o8 = || 107   4376  3187  ||
     || 3897  -4638 2998  ||
     || -1009 2785  -4328 ||
     || 2883  -1540 1031  ||
     || -1517 -713  3879  ||
     +---------------------+
     || 3783  -1437 275   ||
     || 2926  -687  -1310 ||
     || -2582 -2466 1236  ||
     || -2721 -1085 -1127 ||
     || 3072  4259  1727  ||
     +---------------------+
     || -1563 -4868 -1852 ||
     || -968  -960  -1036 ||
     || -3676 382   1454  ||
     || 1795  2177  -4909 ||
     || 2696  -1409 1206  ||
     +---------------------+
```

We may verify that the first row of each matrix in `L` represents a free line, that the second row gives a line through all points, the third rows gives a "pinned line" through the first point, and the fourth and fifth rows give ghost lines through the second and third points, respectively.

```
i9 : netList apply(L,P,(l,c)-> l*c)

     +---------------------+
o9 = || 2418  996   2676 | |
     || 0     0     0    | |
     || 0     -2170 2336 | |
     || -4650 0     4640 | |
     || 4176  -2363 0    | |
     +---------------------+
     || -799  -4310 -4744 ||
     || 0     0     0     ||
     || 0     3153  -4110 ||
     || -1742 0     -2575 ||
     || 2884  -3026 0     ||
     +---------------------+
     || -456  -2638 -1974 ||
     || 0     0     0     ||
     || 0     873   57    ||
     || -2291 0     -422  ||
     || -1987 4218  0     ||
     +---------------------+
```

Now, we define a polynomial ring over FF with indeterminates for the 11 unknown camera parameters:

```
R = FF[r_(1,1), r_(1,2), r_(1,3),
  r_(2,1), r_(2,2), r_(2,3),
  t_(1,1), t_(2,1),t_(1,2), t_(2,2), t_(2,3)]
```

The next line defines a list `Rs` to be a list representing each of the camera matrices' rotations.

```
Rs = {matrix{{1,0,0},{0,1,0},{0,0,1}},
matrix{{ -r_(1,1)^2-r_(1,2)^2+r_(1,3)^2+1,
          -2*r_(1,2)*r_(1,3)+2*r_(1,1),
          2*r_(1,1)*r_(1,3)+2*r_(1,2) },
        { -2*r_(1,2)*r_(1,3)-2*r_(1,1),
          -r_(1,1)^2+r_(1,2)^2-r_(1,3)^2+1,
          -2*r_(1,1)*r_(1,2)+2*r_(1,3) },
        { 2*r_(1,1)*r_(1,3)-2*r_(1,2),
          -2*r_(1,1)*r_(1,2)-2*r_(1,3),
          r_(1,1)^2-r_(1,2)^2-r_(1,3)^2+1 }},
matrix{{ -r_(2,1)^2-r_(2,2)^2+r_(2,3)^2+1,
          -2*r_(2,2)*r_(2,3)+2*r_(2,1),
          2*r_(2,1)*r_(2,3)+2*r_(2,2)},
        { -2*r_(2,2)*r_(2,3)-2*r_(2,1),
          -r_(2,1)^2+r_(2,2)^2-r_(2,3)^2+1,
          -2*r_(2,1)*r_(2,2)+2*r_(2,3) },
        { 2*r_(2,1)*r_(2,3)-2*r_(2,2),
          -2*r_(2,1)*r_(2,2)-2*r_(2,3),
          r_(2,1)^2-r_(2,2)^2-r_(2,3)^2+1 }}};
```

Similarly, we may define a list of camera translations,

```
ts={matrix{{0},{0},{0}},
     matrix{{t_(1,1)},{t_(1,2)},{1}},
     matrix{{t_(2,1)},{t_(2,2)},{t_(2,3)}}};
```

and a list of camera matrices

```
C = apply(Rs,ts,(R,t)->R|t);
```

The polynomials defining each line correspondence (LC) and common point constraint (CP) will depend on `C` and certain submatrices from `L`. The following functions return ideals generated by these polynomials:

```
cl = (C,L') -> minors(3,
        matrix apply(C,L',(c,l)->{l*c}),
        Strategy=>Cofactor);
cp = (C,L') -> minors(4,
        matrix apply(C,L',(c,l)->{l*c}),
        Strategy=>Cofactor);
```

We let `I` be the ideal of camera poses consistent with `L`, which may be constructed as follows:

```
Icl = sum(nVisibleLines,
       i->cl(C,L/(l->l^{i})));
Icp = sum(lineIncidences,
       LINES->cp(C,L/(l->l^LINES)));
I = Icl + Icp;
```

At this point we must note that the matrices defined in `Rs` are "scaled" so that the denominators appearing in the usual Cayley parameterization of $\mathrm{SO}(3)$—namely, $d_1 = r_{2,1}^2 + r_{2,2}^2 + r_{2,3}^2 + 1$ and $d_2 = r_{3,1}^2 + r_{3,2}^2 + r_{3,3}^2 + 1$—do not appear. This scaling has the effect of introducing spurious solutions to the determinantal equations where these denominators may be zero. A standard technique for eradicating these spurious solutions is via introducing an auxiliary variable z and equation $z\,(d_1\,d_2) + 1$ enforcing $d_1 \neq 0$ and $d_2 \neq 0$:

```
Rz = FF[z, gens R,
        MonomialOrder=>{Eliminate 1}]
Iz = sub(I,Rz) + ideal(1+z*
        (r_(2,1)^2+r_(2,2)^2+r_(2,3)^2+1)*
        (r_(1,1)^2+r_(1,2)^2+r_(1,3)^2+1));
```

We may compute a Gröbner basis for the ideal `Iz` using Macaulay2's implementation of the *F4 algorithm.*

```
gbIz = groebnerBasis(Iz, Strategy => "F4");
```

The monomial order for the ring `Rz` is a block-wise graded reverse lexicographical ordering that eliminates the auxiliary variable `z`. It follows [70] that the original ideal `I` is generated by the Gröbner basis elements which do not contain this variable.

```
gbI = selectInSubring(1,gbIz);
```

We get the dimension and degree of `I` by passing to the initial ideal.

```
i23 : inI = ideal sub(leadTerm gbI,R);

o23 : Ideal of R

i24 : dim inI

o24 = 0

i25 : degree inI

o25 = 40
```

C. Computing the Degree Symbolically Using Setup Code

The following Macaulay2 code replicates the entire Gröbner basis computation outlined in the previous section. Note that a seed for the random number generator has been set to ensure a reproducible result.

```
setRandomSeed 0;
m = 3;
FF = ZZ/nextPrime 10000;
isParametric = false;
D = (3,2,{{1,2},{1,3},{1,4}});
needs "problem-builder.m2";
gbIz = groebnerBasis(Iz, Strategy => "F4");
```

The script `"problem-builder.m2"` and its dependencies automate tasks such as building equations and generating point-line data. It assumes several global variables have been defined:

- `m` — the number of cameras
- `FF` — the ground field
- `isParametric` — a Boolean determining how point-line data are represented. A value of `false` indicates that the data are set up randomly over the ground field. A value of `true` means that point-line data are defined in terms of parametric indeterminates, to be specialized later at some fabricated point as in the minimality check described in the next section.
- `D` — an encoding of the problem as described in the previous section.

D. Minimality Check Via Jacobian

We now show how to check minimality without computing degrees for "2111_1."

```
setRandomSeed 0;
m = 3;
FF = ZZ/nextPrime 10000
isParametric = true;
D = (3,2,{{1,2},{1,3},{1,4}});
needs "problem-builder-matrices.m2"
matrices = pointMatrices | lineMatrices;
```

As before, the CP and LC `matrices` are defined over a polynomial ring in 11 indeterminates. However, the coefficient ring is itself a polynomial ring in 27 indetermines, representing the data defining an instance of the problem.

```
i8 : (numgens R, numgens coefficientRing R)

o8 = (11, 27)
```

As in the pseudocode for the minimality check, we fabricate a parameter point satisfying the LC and CP equations:

```
i9 : xy = fabricatedVectorFLPQ D;

              1        38
o9 : Matrix FF  <--- FF

i10 : netList(lineMatrices/(
              m->ker transpose sub(m,xy)))

      +---------------+
o10 = |image | -2428 ||
      |      | -1877 ||
      |      | 1     ||
      +---------------+
      |image | -1975 ||
      |      | 4507  ||
      |      | 1     ||
      +---------------+
      |image | 2657  ||
      |      | -3077 ||
      |      | 1     ||
      +---------------+

i11 : netList(pointMatrices/(
              m->ker sub(m,xy)))

      +---------------+
o11 = |image | -2511 ||
      |      | 417   ||
      |      | -4137 ||
      |      | 1     ||
      +---------------+
      |image | -3306 ||
      |      | 2306  ||
      |      | 1800  ||
      |      | 1     ||
      +---------------+
      |image | 3529  ||
      |      | 2429  ||
      |      | -1474 ||
      |      | 1     ||
      +---------------+
```

The function `goodMinors` then checks minimality by computing a subset of the equations whose Jacobian at `xy` with respective to the 11 camera parameters has maximal rank.

```
i12 : gm = goodMinors(pointMatrices,
                      lineMatrices,xy);

i13 : #gm

o13 : 11
```

E. Computing Degrees Numerically

As an additional check, the degrees of minimal problems with 2 and 3 cameras were re-computed using *monodromy*. [74] This is a general, randomized technique for solving parametric polynomial systems $F(c, y) = 0$ for generic parameter values y. We also used monodromy to compute the degrees of problems with 4 and 5 cameras. For the minimal problem with 6 cameras, monodromy did not terminate after several days, but computed more than 180,000 solutions. Rather than provide detailed pseudocode, we outline the salient features of this approach:

- A generic point $(c_0, y_0) \in \text{Inc}'$ (cf. Section VI) may be sampled as in Algorithm 1. The point then satisfies all line correspondence 6 and common point 7 constraints, written $F(c_0, y_0) = 0$. At any stage of the monodromy computation, known solutions are numerically continued along a fixed set of paths $h_0, \ldots h_j$, where each $h_i : [0, 1] \to \mathcal{Y}_{p,l,\mathcal{I},m}$ is a generic path connecting the solutions

of $F(\cdot, y_0) = 0$ with those of some other fixed instance $F(\cdot, y_1) = 0$. The *solution paths* obtained $\tilde{h_{i,l}} : [0,1] \to \mathcal{C}_m$ satisfy $F(\tilde{h_{i,l}}(t), h_i(t)) = 0$ for all $t \in [0,1]$, and are computed with numerical predictor-corrector methods. Importantly, the paths h_i are *randomized* so as to avoid the singular locus of Inc′ with *probability-one*. Our randomization scheme is analagous to the well-known γ-trick [78] in homotopy continuation.

- Continuation along some path from $t = 0$ to 1 followed by continuation from 1 to 0 along another induces a permutation of the solutions of $F(\cdot, y_0) = 0$. The group of all such permutations acts transitively by the irreducibility of Inc′—thus all solutions may be discovered starting from one. The algorithm maintains a set of correspondences between solutions along each path h_i, and attempts the above continuation step until no other correspondences may be established.
- The stabilization-based stopping criterion previously described comes with no theoretical guarantee of correctness. However, heuristic arguments suggest that all solutions are found with very high probability, even when the number of paths j is small and for problems of moderate-to-large degree. [74], [79] Multiple correct runs for the problems in 2 and 3 views support this thesis, increasing our confidence in the reported results. Nevertheless, we explicitly mark in our table of results the degrees which were only computed numerically with a "*" (cf. [75].)

Computations were performed using the Macaulay2 package `MonodromySolver`. We comment on some technical aspects of these computations:

- To speed up time-intensive polynomial evaluation, polynomials and rational functions appearing in F are implemented as straight-line programs rather than their dense coefficient representations. This makes use of the package `SLPexpressions`.
- It is important to note that the CP and CL constraints determining F result in more equations than unknowns. Taking *all* constraints into account is vital to the correctness of the symbolic computations described in Sections VIII-B and VIII-C. In the monodromy computation, there is more flexibility—we may work with a *square subsystem* with as many equations as unknowns which locally defines the variety Inc′. Provided that the initial point $(c_0, y_0) \in$ Inc′ is generic, all solution curves will remain on Inc′ with probability-one.
- In practice, we deal with numerical approximations to these solution curves, which are subject to the well-known *path-jumping* phenomenon of numerical continuation methods. This means it is possible to compute extraneous solutions satisfying only a square subsystem. To guard against this possibility, the ranks of all CP and CL matrices were computed numerically for each newly discovered solution by thresholding singular values. If any CP or CL matrix did not yield the expected ranks of 3 or 2, the solution was thrown out and not used to generate further correspondences.

The output of the main function `monodromySolve` includes solution data for a chosen parameter point y_i. This solution data may be used as a *start system* for solving an instance of the same problem via homotopy continuation. We illustrate this for the minimal problem "2111_1."

We begin by loading the setup script

```
"numerical-problem-builder.m2".
setRandomSeed 0
m = 3
D = (3,2,{{1,2},{1,3},{1,4}})
Jpivots = {0, 1, 4, 5, 8, 9, 14, 29,
           33, 44, 48, 57, 58, 59}
RERUNMONODROMY = false;
needs "numerical-problem-builder.m2"
```

The global variable `Jpivots` is a list indexing a square subsystem of the LC and CP equations. These equations are the same except that rotations and translations are represented in homogeneous coordinates (i.e. quaternion parameterization of $\mathrm{SO}(3)$)—yielding an extra three variables and equations. The square subsystem, depending on parametric indeterminates $y \in \mathbb{C}^{62}$ as well as $c \in \mathbb{C}^{14}$, is an object of type `GateMatrix` which the setup script assigns to the variable `F`. We may load the results of a previous monodromy run for this problem and verify that the starting solutions are valid:

```
i7 : (yStart, cStarts) =
        readStartSys "2111_1-start";

i8 : max(cStarts/(c->
        norm evaluate(F,c||yStart)))

o8 = 2.49953543586694e-12
o8 : RR (of precision 53)
```

We now consider the problem of reconstructing relative camera poses

```
i13 : netList rotations23

      +-------------------------------+
o13 = || .100226   -.858789 .502431  ||
      || .991267   .129682  .0239204 ||
      || -.0856987 .495646  .864287  ||
      +-------------------------------+
      || -.246694 .614271  .749542  | |
      || .828175  -.268025 .492228  | |
      || .503258  .742182  -.442603 | |
      +-------------------------------+

i14 : netList translations23

      +-------------+
o14 = || .455949   ||
      || .00471457 ||
      || .771746   ||
      +-------------+
      || .74002  |  |
      || .20217  |  |
      || .774737 |  |
      +-------------+
```

from line data in each view

```
i15 : netList L

      +-----------------------------+
o15 = || -.729481 -.373996 .5727   | |
      || -.617925 -.259526 .742169 | |
      || -.33284  -.499815 .799626 | |
      || -.441933 -.298528 .845917 | |
      || -.886353 -.145719 .439481 | |
      +-----------------------------+
      || -.568543 .804717  -.170851 ||
      || -.685669 .639348  -.347982 ||
      || .881415  -.405026 .243023  ||
      || .490952  -.819545 .295486  ||
      || .158324  -.969189 .188695  ||
      +-----------------------------+
      || -.518733 .16617   .838632 | |
      || -.61258  .0842219 .785909 | |
      || -.468687 -.259022 .844535 | |
      || -.595602 -.232359 .768939 | |
      || -.589568 .295501  .751724 | |
      +-----------------------------+.
```

The derived (parameter, solution) pair is defined by respective variables `yTarget`, `cTarget`. The starting solutions are numerically continued to solutions of `F` specialized at `yTarget` using core functions for `NumericalAlgebraicGeometry` in Macaulay2.

```
PH' = specialize(PH,yStart||yTarget);
cTargets = trackHomotopy(PH', cStarts);
```

The following code filters the target solutions `cTargets` for the ground-truth solution `y`.

```
cTargets2 = cTargets/(c-> (
    R2params = take(c.Coordinates,{0,3});
    Q2R(R2params,Normalized=>true)));
i=minPosition(
    cTargets2/(R2->
    norm(R2-rotations23#0)));
y=(cTargets#i).Coordinates;
```

We recover the original camera poses:

```
i23 : Q2R(take(y,{0,3}),Normalized=>true)

o23 = | .100226   -.858789 .502431  |
      | .991267   .129682  .0239204 |
      | -.0856987 .495646  .864287  |

                  3          3
o23 : Matrix CC    <--- CC
               53         53

i24 : Q2R(take(y,{4,7}),Normalized=>true)

o24 = | -.246694 .614271  .749542  |
      | .828175  -.268025 .492228  |
      | .503258  .742182  -.442603 |

                  3          3
o24 : Matrix CC    <--- CC
               53         53

i25 : transpose matrix take(y,{8,13})

o25 = | .455949   |
      | .00471457 |
      | .771746   |
      | .74002    |
      | .20217    |
      | .774737   |
```

IX. Conclusion

Our work characterizes a new class of minimal problems and provides a complete classification of generic minimal problems for complete visibility with calibrated cameras. One clear direction for future work is to extend our results for the the fully calibrated scenario towards partial calibration. It is also natural to relax the assumption of complete visibility. In this direction, the case of three views has already been addressed in follow-up work [69].

Only a few of the cases appearing our classification had been discovered prior to this work. Several new cases possess a small number of solutions and hence call for constructing efficient solvers [3], [17]. Two of the harder problems in our classification, namely 3010_0 and 3002_1, have already been implemented considerably more efficiently using homotopy continuation solvers [80], reaching sub-second run times. Despite the fact that many of the degrees appearing in Table I are quite large, recent work has showed that some of these problems can be decomposed into sub-problems, allowing for more efficient solutions; this idea has recently been successfully pursued in the context of other minimal problems [81]. On the other hand, for problems with unavoidably large degrees, learned homotopy continuation-based heuristics developed in [82] might still provide efficient practical solutions. With continued advances in matching and detecting line features in images, eg. [83], [84], there is ample motivation for further study of the problems introduced here, towards their eventual use in practical structure from motion pipelines.

Acknowledgment

We thank the many research visitors at ICERM for fruitful discussions on minimal problems.

References

[1] 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.

[2] H. Stewenius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS J. Photogrammetry Remote Sens.*, vol. 60, pp. 284–294, 2006.

[3] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *Proc. Eur. Conf. Comput. Vis.*, 2008.

[4] M. Byröd, K. Josephson, and K. Åström, "A column-pivoting based strategy for monomial ordering in numerical Gröbner basis calculations," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 130–143.

[5] S. Ramalingam and P. F. Sturm, "Minimal solutions for generic imaging models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[6] A. Elqursh and A. M. Elgammal, "Line-based relative pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3049–3056.

[7] M. F. Mirzaei and S. I. Roumeliotis, "Optimal estimation of vanishing points in a Manhattan world," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2454–2461.

[8] L. Kneip, R. Siegwart, and M. Pollefeys, "Finding the exact rotation between two images independently of the translation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 696–709.

[9] R. Hartley and H. Li, "An efficient hidden variable approach to minimal-case camera motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2303–2314, Dec. 2012.

[10] Y. Kuang and K. Åström, "Stratified sensor network self-calibration from TDOA measurements," in *Proc. 21st Eur. Signal Process. Conf.*, 2013, pp. 1–5.

[11] 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.

[12] O. Saurer, M. Pollefeys, and G. H. Lee, "A minimal solution to the rolling shutter pose estimation problem," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2015, pp. 1328–1334.

[13] J. Ventura, C. Arth, and V. Lepetit, "An efficient minimal solution for multi-camera motion," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 747–755.

[14] 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.

[15] 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.

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

[17] V. Larsson, K. Åström, and M. Oskarsson, "Polynomial solvers for saturated ideals," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2307–2316.

[18] V. Larsson, Z. Kukelova, and Y. Zheng, "Making minimal solvers for absolute pose estimation compact and robust," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2335–2343.

[19] Z. Kukelova, J. Kileel, B. Sturmfels, and T. Pajdla, "A clever elimination strategy for efficient minimal solvers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3605–3614.

[20] V. Larsson, M.K. Oskarsson, A. Åström, Z. W. Kukelova, and T. Pajdla, "Beyond Gröbner bases: Basis selection for minimal solvers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3945–3954.

[21] S. Agarwal, H.-L. Lee, B. Sturmfels, and R. R. Thomas, "On the existence of epipolar matrices," *Int. J. Comput. Vis.*, vol. 121, no. 3, pp. 403–415, 2017.

[22] D. Barath, T. Toth, and L. Hajder, "A minimal solution for two-view focal-length estimation using two affine correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2557–2565.

[23] D. Barath, "Five-point fundamental matrix estimation for uncalibrated cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 235–243.

[24] D. Barath and L. Hajder, "Efficient recovery of essential matrix from two affine correspondences," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5328–5337, Nov. 2018.

[25] P. Miraldo, T. Dias, and S. Ramalingam, "A minimal closed-form solution for multi-perspective pose estimation using points and lines," in *Proc. Comput. Vis. 15th Eur. Conf.*, 2018, pp. 490–507.

[26] V. Larsson, T.Z. S. Kukelova, and M. Pollefeys, "Revisiting radial distortion absolute pose," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1062–1071.

[27] S. Bhayani, Z. Kukelova, and J. Heikkila, "A sparse resultant based method for efficient minimal solvers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1767–1776.

[28] A. Mateus, S. Ramalingam, and P. Miraldo, "Minimal solvers for 3D scan alignment with pairs of intersecting lines," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7232–7242.

[29] Y. Ding, J. Yang, J. Ponce, and H. Kong, "Minimal solutions to relative pose estimation from two views sharing a common direction with unknown focal length," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7043–7051.

[30] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," in *Proc. ACM Siggraph Papers*, 2006, pp. 835–846.

[31] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 189–210, 2008.

[32] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[33] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1658–1669.

[34] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. Comput. Vis. Pattern Recognit.*, 2004, pp. 652–659.

[35] H.S. Alismail, B. Browning, and M. B. Dias, "Evaluating pose estimation methods for stereo visual odometry on robots," in *Proc. 11th Int. Conf. Intell. Auton. Syst.*, 2011, pp. 302–315.

[36] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1744–1756, Sep. 2017.

[37] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1455–1461, Jul. 2017.

[38] H. Taira et al., "InLoc: Indoor visual localization with dense matching and view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7199–7209.

[39] T. Schöps et al., "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3260–3269.

[40] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2003.

[41] D. OlivierFaugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 2, no. 3, pp. 485–508, 1988.

[42] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control," Tech. Rep. 6303, INRIA, 2007.

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

[44] Hugh Christopher Longuet-Higgins, "A method of obtaining the relative positions of four points from three perspective projections," *Image Vis. Comput.*, vol. 10, no. 5, pp. 266–270, 1992.

[45] J. RobertHolt and A. N. Netravali, "Uniqueness of solutions to three perspective views of four points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 3, pp. 303–307, Mar. 1995.

[46] 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.

[47] 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.

[48] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[49] J. Matas, S. Obdrzálek, and O. Chum, "Local affine frames for wide-baseline stereo," in *Proc. 16th Int. Conf. Pattern Recognit.*, 2002, pp. 363–366.

[50] B. Johansson, M. Oskarsson, and K. Åström, "Structure and motion estimation from complex features in three views," in *Proc. 3rd Indian Conf. Comput. Vis. Graph. Image Process.*, 2002.

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

[52] D. Nistér, R. I. Hartley, and H. Stewénius, "Using Galois theory to prove structure from motion algorithms are optimal," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[53] T. Duff, V. Korotynskiy, T. Pajdla, and M. H. Regan, "Galois/monodromy groups for decomposing minimal problems in 3D reconstruction," *SIAM J. Appl. Algebra Geometry*, vol. 6, no. 4, pp. 740–772, 2022.

[54] T. Duff, K. Kohn, A. Leykin, and T. Pajdla, "PLMP - point-line minimal problems in complete multi-view visibility," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 1675–1684.

[55] Richard I. Hartley, "Lines and points in three views and the trifocal tensor," *Int. J. Comput. Vis.*, vol. 22, no. 2, pp. 125–140, 1997.

[56] Y. Ma, K. Huang, R. Vidal, J. Kosecka, and S. Sastry, "Rank conditions on the multiple-view matrix," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 115–137, 2004.

[57] 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, pp. 2383–2392.

[58] R. Fabbri, P. J. Giblin, and B. B. Kimia, "Camera pose estimation using first-order curve differential geometry," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 231–244.

[59] R. Fabbri and B. B. Kimia, "Multiview differential geometry of curves," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 324–346, 2016.

[60] 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.

[61] D. Barath and J. Matas, "Progressive-X: Efficient, anytime, multi-model fitting algorithm," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3779–3787.

[62] B. Guan, J.Z. Zhao, F. L. Sun, and F. Fraundorfer, "Minimal solutions for relative pose with a single affine correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1926–1935.

[63] R. Raguram, C. Wu, J. -M. Frahm, and S. Lazebnik, "Modeling and recognition of landmark image collections using iconic scene graphs," *Int. J. Comput. Vis.*, vol. 95, no. 3, pp. 213–239, 2011.

[64] M. Joswig, J. Kileel, B. Sturmfels, and A. Wagner, "Rigid multiview varieties," in *Proc. Int. J. Algebra Comput.*, vol. 26, no. 4, pp. 775–788, 2016.

[65] C. Aholt and L. Oeding, "The ideal of the trifocal variety," *Math. Comput.*, vol. 83, no. 289, pp. 2553–2574, 2014.

[66] C. Aholt, B. Sturmfels, and R. Thomas, "A Hilbert scheme in computer vision," *Can. J. Math.*, vol. 65, no. 5, pp. 961–988, 2013.
[67] M. Trager, "Cameras, shapes, and contours: Geometric models in computer vision. (caméras, formes et contours: Modèles géométriques en vision par ordinateur). PhD thesis, École Normale Supérieure, Paris, France, 2018.
[68] M. Trager, J. Ponce, and M. Hebert, "Trinocular geometry revisited," *Int. J. Comput. Vis.*, vol. 120, pp. 134–152, 2016.
[69] T. Duff, K. Kohn, A. Leykin, and T. Pajdla, "Pl $_1$ p - point-line minimal problems under partial visibility in three views," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 175–192.
[70] 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.
[71] B. Triggs, "Matching constraints and the joint image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1995, pp. 338–343.
[72] M. Trager, M. Hebert, and J. Ponce, "The joint image handbook," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 909–917.
[73] R. D. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry," [Online]. Available: http://www.math.uiuc.edu/Macaulay2/
[74] 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.*, pp. 1421–1446, 2019.
[75] J. D. Hauenstein, L. Oeding, G. Ottaviani, and A. J. Sommese, "Homotopy techniques for tensor decomposition and perfect identifiability," *J. Die Reine Angewandte Mathematik*, 753, pp. 1–22, 2019.
[76] K. Kohn, B. Shapiro, and B. Sturmfels, "Moment varieties of measures on polytopes," *Annali Della Scuola Normale Superiore Di Pisa*, vol. 21, pp. 739–770, 2020.
[77] P. Breiding, B. Sturmfels, and S. Timme, "3264 conics in a second," 2019, *arXiv: 1902.05518*.
[78] A. Morgan and A. Sommese, "A homotopy for solving general polynomial systems that respects M-homogeneous structures," *Appl. Math. Comput.*, vol. 24, no. 2, pp. 101–113, 1987.
[79] N. Bliss, T. Duff, A. Leykin, and J. Sommars, "Monodromy solver: Sequential and parallel," in *Proc. ACM Int. Symp. Symbolic Algebr. Comput.*, 2018, pp. 87–94.
[80] R. Fabbri et al., "TRPLP - trifocal relative pose from lines at points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12070–12080.
[81] P. Hruby et al., "Four-view geometry with unknown radial distortion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8990–9000.
[82] 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.
[83] R. Pautrat, I. Suárez, Y. Yu, M. Pollefeys, and V. Larsson, "GlueStick: Robust image matching by sticking points and lines together," in *Proc. Int. Conf. Comput. Vis.*, 2023.
[84] R. Pautrat, D. Barath, V. Larsson, M. R. Oswald, and M. Pollefeys, "DeepLSD: Line segment detection and refinement with deep image gradients," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17327–17336.

Timothy Duff received the PhD degree from the Algorithms, Combinatorics, and Optimization, Georgia Tech. He is a NSF postdoctoral scholar and postdoctoral fellow at the University of Washington in Seattle. His research interests include computational/combinatorial aspects of algebraic geometry and their application in fields like computer vision. He is a member of the AMS and SIAM.

Kathlén Kohn received PhD degree from the Technical University of Berlin. She is a WASP-AI/Math assistant professor with the KTH Royal Institute of Technology in Stockholm. She works in nonlinear algebra with a focus on applications in data science and artificial intelligence. Her recent work concerns maximum likelihood estimation in statistics, the theoretical foundations of machine learning with neural networks, and 3D reconstruction in computer vision.

Anton Leykin received the PhD degree from the University of Minnesota, in 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 of resulting algorithms in Macaulay2 computer algebra system. He is a member of the ACM, AMS, and SIAM.

Tomas Pajdla (Member, IEEE) received the PhD degree from Czech Technical University, in Prague. He works in geometry and algebra of computer vision and robotics with emphasis on minimal problems, 3D reconstruction, and industrial vision. He contributed to introducing epipolar geometry of panoramic cameras, non-central camera models generated by linear mapping, generalized epipolar geometries, to developing solvers for minimal problems in structure from motion and to solving image matching problem. He coauthored works awarded prizes at OAGM 1998 and 2013, BMVC 2002, ACCV 2014, and ICCV 2019.