Theoretical and Numerical Analysis of 3D Reconstruction Using Point and Line Incidences

Felix Rydell
KTH Royal Institute of Technology
Lindstedtsvägen 25, Stockholm, Sweden
felixry@kth.se

Elima Shehu
University of Osnabrück and MPI MiS
Albrechtstrasse 28a, Osnabrück, Germany
Inselstrasse 22, Leipzig, Germany
elima.shehu@mis.mpg.de

Angélica Torres
Centre de Recerca Matemàtica
Edifici C, Campus UAB, Bellaterra, Spain
atorres@crm.cat

Abstract

We study the joint image of lines incident to points, meaning the set of image tuples obtained from fixed cameras observing a varying 3D point-line incidence. We prove a formula for the number of complex critical points of the triangulation problem that aims to compute a 3D point-line incidence from noisy images. Our formula works for an arbitrary number of images and measures the intrinsic difficulty of this triangulation. Additionally, we conduct numerical experiments using homotopy continuation methods, comparing different approaches of triangulation of such incidences. In our setup, exploiting the incidence relations gives a notably faster point reconstruction with comparable accuracy.

Introduction

The Structure-from-Motion pipeline aims to estimate the camera positions and the relative position of the objects present in a given set of images. The process starts by identifying point and line features in one image that are recognizable as the same points or lines in another. These matched features, that we call *correspondences*, are used to estimate the camera position and orientation, which then allow for the triangulation step where the position of the points and lines is estimated.

In this work, we study the triangulation of points and lines satisfying a certain incidence relation. Specifically, we focus on the triangulation of points contained in a single line; see Figure 1. We note that our methods can also be used when multiple lines going through a fixed point. We develop the theory for the triangulation of these problems for an arbitrary number of pinhole cameras assuming complete visibility.

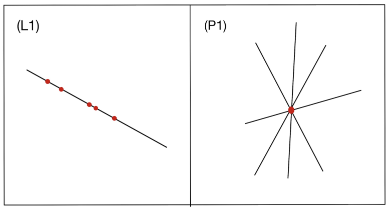

Figure 1: The illustration of two different types of incidence relations: (L1) represents the scenario where multiple points are incident to a line, while (P1) represents the scenario where multiple lines are incident to a point.

The triangulation of points is well understood and, in practice, it is efficiently implemented. However, to the best of our knowledge, incidence relations are not considered in these implementations. The inclusion of line features and incidence relations in the triangulation process could give a more accurate estimation of scenes as lines are more robust to noise than points, and incidence relations appear frequently in interior scenes and man-made scenarios; see [34], [24], [30]. Additionally, in some real data sets standard feature detection algorithms fail due to a lack of point correspondences but succeed when line correspondences are taken into account [15].

The main tools in our work are *(algebraic) varieties*, that is, the vanishing sets of systems of polynomial equations. Algebraic varieties have been used extensively to study triangulation of point correspondences [1, 17, 31], line correspondences [7, 16, 29] and minimal problems [12, 13] to name a few. In these works, algebraic varieties arise naturally due to the algebraic nature of pinhole cameras.

A *pinhole camera*, is modeled by a (complex) projective linear map $C : \mathbb{P}^3 \dashrightarrow \mathbb{P}^2$ defined by a 3×4 matrix C of full rank that takes a point $X \in \mathbb{P}^3$ and sends it to $CX \in \mathbb{P}^2$. A camera arrangement with $m \geq 2$ cameras is denoted by $\mathcal{C} = (C_1, \dots, C_m)$, and the *joint camera map*

$$\begin{aligned} \Phi_{\mathcal{C}} : \mathbb{P}^3 &\dashrightarrow (\mathbb{P}^2)^m, \\ X &\mapsto (C_1X, \dots, C_mX), \end{aligned} \tag{1}$$

models the process of taking the images of a point X in homogeneous coordinates with m cameras. For fixed cameras, the *(point) multiview variety* $\mathcal{M}_{\mathcal{C}}$ is the smallest variety that contains all point correspondences. An extensive account of the pinhole cameras is given by [22], and a survey of the multiview variety is found in [36]. The joint camera map can be extended from the space of points $\mathbb{P}^3$ to the space of lines, denoted by $\mathrm{Gr}(1, \mathbb{P}^3)$, where each line is parametrized as the span of two points. The map

$$\begin{aligned} \Upsilon_{\mathcal{C}} : \mathrm{Gr}(1, \mathbb{P}^3) &\dashrightarrow (\mathbb{P}^2)^m, \\ L &\mapsto (C_1 \cdot L, \dots, C_m \cdot L). \end{aligned} \tag{2}$$

models the image of a line L taken by the cameras of $\mathcal{C}$. To be precise, if u, v span the line L in $\mathbb{P}^3$, then $C \cdot L$ is defined as $\ell = Cu \times Cv \in \mathbb{P}^2$, where $\times$ is the cross product. Observe that $\{x \in \mathbb{P}^2 : \ell^T x = 0\}$ equals $\mathrm{span}\{Cu, Cv\}$. Recently, in [7], the authors study the *line multiview variety* referred to as $\mathcal{L}_{\mathcal{C}}$, which is the smallest variety containing all line correspondences.

Our main contribution is the definition and study of the *anchored* point and line multiview varieties. Given a fixed line L in $\mathbb{P}^3$, the anchored point multiview variety, $\mathcal{M}_{\mathcal{C}}^L$, is defined as the smallest variety containing all point correspondences coming from points in L; and the anchored line multiview variety, $\mathcal{L}_{\mathcal{C}}^X$, is the smallest variety containing the line correspondences coming from lines passing through a given point $X \in \mathbb{P}^3$.

For these new varieties we prove formulas for their *Euclidean distance degree* (EDD), which are a measurement of the complexity for error correction using exact algebraic methods [11]. Specifically, we prove the following theorem:

Theorem 0.1. *Let $\mathcal{C}$ be a generic arrangement of m cameras.*

1. $\mathrm{EDD}(\mathcal{M}_{\mathcal{C}}^L) = 3m - 2$.
2. *If $m \geq 3$, then* $\mathrm{EDD}(\mathcal{L}_{\mathcal{C}}^X) = \frac{9}{2}m^2 - \frac{19}{2}m + 3$.

We provide a precise definition of this degree in Section 1.2. Previous work on EDD for multiview varieties includes [19, 31] and [7, Section 5]. The EDD of the point multiview variety is $\mathrm{EDD}(\mathcal{M}_{\mathcal{C}}) = \frac{9}{2}m^3 - \frac{21}{2}m^2 + 8m - 4$, according to [31]. The fact that the EDDs of the anchored multiview varieties are polynomials of smaller degrees suggests that they are less complex for the purpose of data correction. This conclusion is backed by the results of our numerical experiments using `HomotopyContinuation.jl` [9], presented in Section 2.

Another contribution of our work is the numerical simulations for the triangulation of points contained in a line using the anchored multiview varieties. We present different approaches to triangulating data of type (L1) that differ from the traditional method of fitting point correspondences to the multiview variety $\mathcal{M}_{\mathcal{C}}$. For $m = 2$ views, our implementation is notably faster than the traditional triangulation of points described above, while the accuracy is comparable. For $m = 3$ views we get both higher accuracy and faster speed by using $\mathcal{L}_{\mathcal{C}}$, compared to usual point triangulation. All of our proposed methods outperform the traditional approaches in terms of run-time and give a comparably accurate result. In practice, special software and hardware are used for triangulation, and based on our experiments, we believe that these approaches could be implemented efficiently with good results.

Related work

Algebra. Algebraic geometry, whose connection to computer vision is well established, is our main tool for the theoretical study of the triangulation of problem (L1). Fundamental theory regarding the point multiview variety is found in [17, 20, 22], in particular for two and three views. The paper [36] by Trager et. al. serves as a survey for the algebraic properties of this multiview variety. For applications, finding a good set of polynomial constraints satisfied by the point correspondences is important, this is precisely the work of [1] by Agarwal et al.

Regarding the algebra of lines in a computer vision setting, Kileel presented in [29, Definition 3.9, Theorem 3.10] several types of multiview varieties with respect to a point-line incidence relation in $\mathbb{P}^3$ in 3 views, and their basic properties. One of those types is called the LLL-multiview variety and constitutes a special case of the more recent work by Breiding et. al. [7], where they study algebraic properties of the line multiview variety. Furthermore, a recent manuscript [6] studies the polynomial constraints satisfied by line correspondences.

Projective Reconstruction. Different approaches to the triangulation of points have been considered in the literature, in particular for two views [3, 4, 21, 26–28]. Hartley and Sturm compare many different approaches in [21], including the “midpoint” method. The midpoint method inputs a point correspondence in two views and outputs the midpoint of the line segment determined by the points on the two back-projected lines that are closest to each other. In other words, it finds the midpoint of the common perpendicular to the two back-projected lines. This is a natural approach, but it has several downsides, also pointed out in [3, 4]. Importantly, it is not *projectively invariant*. A projectively invariant reconstruction has the property that acting on the cameras by a global projective transformation induces the same action on the triangulated point or line. Instead, Hartley and Sturm propose minimizing the reprojection error as the optimal triangulation method, which is accepted as a standard formulation [35] and it is projectively invariant. In [35], the focus is on triangulation via minimizing reprojections errors in three views and the authors point out that three views often lead to greater stability and stronger disambiguation compared to two views.

Implementation of Line Reconstruction. From the algorithmic point of view, the simultaneous reconstruction of point and line features have been studied specially with the goal of reconstructing line segments. For example, [2] provides a thorough overview of the structure-from-motion pipeline using lines, going through different methods for line parameterization and error correction depending on such parameterizations. In [33], Quan and Takeo study the algebraic structure of line correspondences with uncalibrated affine cameras. They reduce the problem of reconstructing affine lines to the reconstruction of projective points in a lower-dimensional projective camera. In [20], Hartley and coauthors provide an algorithm for the reconstruction of point and line features where 3D lines are parameterized by their projections in 2 views (as the intersection of two back-projected planes). Micusik and Wildenauer reconstruct lines to estimate line segments, but only use incident points for error correction [32]. Furthermore in [23], the authors propose a technique for 3D reconstruction incorporating line segments. They assert that this method surpasses traditional approaches, especially in efficiency while getting accurate results. Finally, in [14], the authors present a framework for the computation of the relative motion between two images using a triplet of lines.

This paper is structured as follows. In the first part of Section 1 we formally introduce anchored multiview varieties and study some properties. In the second part of Section 1, we study their smoothness and find their Euclidean Distance Degree. Finally, in Section 2, we provide a numerical analysis of different approaches to reconstructing point correspondences incident to a line correspondence. The proofs of all our results are included in the Supplementary Material together with a small background on the concepts of Algebraic Geometry used for these results and the code for the numerical experiments.

1. Anchored Multiview Varieties

A *camera* refers to a full-rank 3×4 matrix. A *camera arrangement* is a collection $\mathcal{C} = (C_1, \dots, C_m)$ of $m \geq 2$ cameras whose *centers*, i.e. kernels, are all distinct.

A *variety* is the solution set to a system of polynomial equations. The *Zariski closure* of a set U is the smallest variety containing U. We work in complex projective space, denoted $\mathbb{P}^n$. This is defined as $(\mathbb{C}^n \setminus \{0\})/\sim$, where $x \sim y$ if x and y differ by a non-zero constant. The set of lines in $\mathbb{P}^n$ is denoted as $\mathrm{Gr}(1, \mathbb{P}^n)$. In the language of algebraic geometry, it is called the *Grassmanian of lines* in $\mathbb{P}^n$. In general we denote with upper case letters world objects and with lower case letters image objects. For a rational map $\varphi : X \dashrightarrow Y$, we denote $\varphi|_A$ the restriction of φ to $A \subseteq X$.

Definition 1.1. *Let $X \in \mathbb{P}^3$ be a point distinct from all camera centers, and L a line in $\mathbb{P}^3$ containing none of the camera centers.*

1. *The* anchored point multiview variety, *denoted $\mathcal{M}_{\mathcal{C}}^L$, is defined as the Zariski closure of the image of the map*

$$\begin{aligned} \Phi_{\mathcal{C}}|_L : L &\dashrightarrow (\mathbb{P}^2)^m, \\ X &\longmapsto (C_1 X, \dots, C_m X). \end{aligned} \tag{3}$$

2. *The* anchored line multiview variety, *denoted $\mathcal{L}_{\mathcal{C}}^X$, is defined as the Zariski closure of the image of the map*

$$\begin{aligned} \Upsilon_{\mathcal{C}}|_{\Lambda(X)} : \Lambda(X) &\dashrightarrow (\mathbb{P}^2)^m, \\ L &\longmapsto (C_1 \cdot L, \dots, C_m \cdot L), \end{aligned} \tag{4}$$

where $\Lambda(X)$ denotes the set of lines in $\mathbb{P}^3$ that contain X and $\ell_i = C_i \cdot L$ is defined as in the introduction.

The anchored point multiview variety $\mathcal{M}_{\mathcal{C}}^L$ is the smallest variety that contains all point correspondences $\Phi_{\mathcal{C}}(X)$ for $X \in L$. Similarly, the anchored line multiview variety $\mathcal{L}_{\mathcal{C}}^X$ is the smallest variety that contains all line correspondences $\Upsilon_{\mathcal{C}}(L)$ for L meeting X and no center. We highlight that our definition of anchored multiview varieties is different from the anchored features in [25] for monocular EKF-SLAM.

Proposition 1.2. *Consider an arrangement of m cameras $\mathcal{C} = (C_1, \dots, C_m)$, a point $X \in \mathbb{P}^3$ and a line L in $\mathbb{P}^3$ satisfying the conditions of Definition 1.1.*

1. *If there are two different camera centers c_i and c_j such that the span of $\{c_i, c_j, L\}$ is $\mathbb{P}^3$, then*

$$\mathcal{M}_{\mathcal{C}}^L = \{(x_1, \dots, x_m) \in \mathcal{M}_{\mathcal{C}} : x_i \in C_i \cdot L\}. \tag{5}$$

2. *If for each camera center c_i, the line spanned by c_i and X does not contain any other camera center, then*

$$\mathcal{L}_{\mathcal{C}}^X = \{(\ell_1, \dots, \ell_m) \in \mathcal{L}_{\mathcal{C}} : C_i X \in \ell_i\}. \tag{6}$$

The proofs for this and all our results are found in the Supplementary Material.

1.1. Properties of Anchored Multiview Varieties

A fundamental property of the anchored multiview varieties is that they are linearly isomorphic to multiview varieties arising from projections $\mathbb{P}^1 \dashrightarrow \mathbb{P}^1$ and $\mathbb{P}^2 \dashrightarrow \mathbb{P}^1$, respectively. A linear isomorphism is a linear map (given by a matrix) with an inverse that is also linear.

Consider an arrangement $\widetilde{\mathcal{C}}$ of m full-rank 2×2 matrices, and an arrangement $\widehat{\mathcal{C}}$ of m full-rank 2×3 matrices. Also here we assume $m \geq 2$. We define $\mathcal{M}^{1,1}_{\widetilde{\mathcal{C}}}$, and $\mathcal{M}^{2,1}_{\widehat{\mathcal{C}}}$, respectively as the Zariski closure of the image of the maps

$$\begin{aligned} \Phi_{\widetilde{\mathcal{C}}} : \mathbb{P}^1 &\longrightarrow (\mathbb{P}^1)^m, \\ X &\longmapsto (\widetilde{C}_1 X, \dots, \widetilde{C}_m X) \end{aligned} \tag{7}$$

and

$$\begin{aligned} \Phi_{\widehat{\mathcal{C}}} : \mathbb{P}^2 &\dashrightarrow (\mathbb{P}^1)^m, \\ X &\longmapsto (\widehat{C}_1 X, \dots \widehat{C}_m X). \end{aligned} \tag{8}$$

Theorem 1.3.

1. *Let $\phi_L : L \to \mathbb{P}^1$ and $\psi_{\mathcal{C},i} : C_i \cdot L \to \mathbb{P}^1$ be any choices of linear isomorphisms. Let $\widetilde{\mathcal{C}}$ denote the arrangement of matrices $\widetilde{C}_i := \psi_{\mathcal{C},i} \circ C_i \circ \phi_L^{-1}$. Then*

$$\psi_{\mathcal{C},L} := (\psi_{\mathcal{C},1}, \dots, \psi_{\mathcal{C},m}) : \mathcal{M}^L_{\mathcal{C}} \to \mathcal{M}^{1,1}_{\widetilde{\mathcal{C}}} \tag{9}$$

is a linear isomorphism.

2. *Let $\phi_X : \Lambda(X) \to \mathbb{P}^2$ and $\psi_{\mathcal{C},i} : \Lambda(C_i X) \to \mathbb{P}^1$ be any choices of linear isomorphisms. Let $\widehat{\mathcal{C}}$ denote the arrangement of matrices $\widehat{C}_i := \psi_{\mathcal{C},i} \circ C_i \circ \phi_X^{-1}$. Then*

$$\psi_{\mathcal{C},X} := (\psi_{\mathcal{C},1}, \dots, \psi_{\mathcal{C},m}) : \mathcal{L}^X_{\mathcal{C}} \to \mathcal{M}^{2,1}_{\widehat{\mathcal{C}}} \tag{10}$$

is a linear isomorphism.

In Section 2.1.1, we exploit these linear isomorphisms to improve the speed of triangulation for our setting. A consequence of this theorem is also that many results for the anchored multiview varieties translate into results about $\mathcal{M}^{1,1}_{\widetilde{\mathcal{C}}}$ and $\mathcal{M}^{2,1}_{\widehat{\mathcal{C}}}$.

Proposition 1.4. *$\mathcal{M}^L_{\mathcal{C}}$ and $\mathcal{L}^X_{\mathcal{C}}$ are irreducible. Further,*

1. *$\mathcal{M}^L_{\mathcal{C}}$ is isomorphic to $\mathbb{P}^1$. In particular, $\dim \mathcal{M}^L_{\mathcal{C}} = 1$.*
2. *If the span of the centers c_i and the point X are not collinear, then $\dim \mathcal{L}^X_{\mathcal{C}} = 2$.*

Using the fundamental matrices of C_i and C_j, denoted as F^{ij}, we introduce a set of polynomial constraints that must be fulfilled by the correspondences of points or lines in the anchored multiview varieties. These constraints imply that some of the equations obtained from Proposition 1.2 are redundant in the generic case.

Proposition 1.5. *For a point $X \in \mathbb{P}^3$ and line L in $\mathbb{P}^3$, let $\mathcal{C}$ be a generic (random) camera arrangement of m cameras.*

1. *$x \in \mathcal{M}^L_{\mathcal{C}}$ if and only if $x_1^T F^{1j} x_j = 0$ for every $j = 2, \dots, m$ and $x_i^T C_i \cdot L = 0$ for every $i = 1, \dots, m$.*
2. *$\ell \in \mathcal{L}^X_{\mathcal{C}}$ if and only if*

$$\begin{aligned} \det \begin{bmatrix} C_1^T \ell_1 & C_2^T \ell_2 & C_i^T \ell_i \end{bmatrix} &= 0, \\ \det \begin{bmatrix} C_1^T \ell_1 & C_3^T \ell_3 & C_i^T \ell_i \end{bmatrix} &= 0 \end{aligned} \tag{11}$$

for $i = 3, \dots, m$ and $\ell_i^T C_i X = 0$ for every $i = 1, \dots, m$.

The determinantal constraints described in the second item correspond to the constraints that are satisfied by elements of the line multiview variety, presented in [7].

In Supplementary Material Section B we define the *multidegree* of a variety in a product of projective spaces, determine it for the anchored multiview varieties and explain its relevance for computer vision.

1.2. The Euclidean Distance problem

For a variety $\mathcal{X} \subseteq \mathbb{R}^n$ and a point $u \in \mathbb{R}^n$ outside the variety, a natural problem is to find the closest point on $\mathcal{X}$ to u, which corresponds to the optimization problem

$$\text{minimize} \quad \sum_{i=1}^{n} (u_i - x_i)^2 \quad \text{subject to} \quad x \in \mathcal{X}. \tag{12}$$

Equation (12) is called the *Euclidean distance problem* and models the process of error correction and fitting noisy data to a mathematical model $\mathcal{X}$. In the case of a smooth variety, defined below, the *Euclidean distance degree* (EDD) is the number of complex solutions to the critical equations of (12) [11]. The EDD is an estimate of how difficult it is to solve this problem by exact algebraic methods. For a variety $\mathcal{X}$ in a product of projective spaces $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_m}$, the EDD is the EDD of $\mathcal{X} \cap U_1 \times \cdots \times U_m \subseteq \mathbb{R}^{n_1 + \cdots + n_m}$, where U_i is a generic affine patch of $\mathbb{P}^{h_i}$ for each i. An *affine patch* U of $\mathbb{P}^n$ is a subset defined by an affine equation with non-zero constant part, for instance $x_0 = 1$. We have $U \cong \mathbb{R}^n$ over the real numbers.

A variety $\mathcal{X}$ is *smooth* at a point x if $\mathcal{X}$ locally around x looks like Euclidean space. $\mathcal{X}$ is *smooth* if all its points are smooth. The *singular locus* of a variety is the set of non-smooth points. See [18] for more details. The singular locus of the point multiview variety is well-understood, see for instance [36], and it is mostly understood for the line multiview variety, see [7, Section 3]. The proposition below guarantees that the anchored multiview varieties are smooth for generic camera arrangements, which helps us to compute its EDD.

Proposition 1.6.

1. *$\mathcal{M}_{\mathcal{C}}^{L}$ smooth.*
2. *If there are exactly two cameras, or the centers together with the point X span $\mathbb{P}^3$, then $\mathcal{L}_{\mathcal{C}}^{X}$ is smooth.*

For the EDD to be relevant in a particular setting, the Euclidean distance has to be a good measurement of distance. This is naturally the case for points in $\mathbb{R}^2$, i.e. an affine patch of $\mathbb{P}^2$. This extends to an affine patch of $\mathcal{M}_{\mathcal{C}}$. It is perhaps less clear how to best measure distances between lines. Recall that we regard $\mathcal{L}_{\mathcal{C}}$ as a subset of $(\mathbb{P}^2)^m$. Indeed, we identify image lines with the point that defines its normal vector. We choose the Euclidean distance between these normal vectors (in affine patches of $\mathbb{P}^2$) as our distance between lines.

The theorem below present our computations of the EDDs of the anchored multiview varieties. Additionally, our numerical computations using the `HomotopyContinuation.jl` [8] package in the `julia` [5] programming language have confirmed the accuracy of these formulas for $m \leq 10$.

Theorem 1.7. *Let $\mathcal{C}$ be a generic arrangement of m cameras.*

1. $\mathrm{EDD}(\mathcal{M}_{\mathcal{C}}^{L}) = 3m - 2$.
2. *If $m \geq 3$, then* $\mathrm{EDD}(\mathcal{L}_{\mathcal{C}}^{X}) = \frac{9}{2}m^2 - \frac{19}{2}m + 3$.

We end with an implication of Theorem 1.3: There is a direct correspondence between the EDDs of the anchored multiview varieties and $\mathcal{M}_{\widetilde{\mathcal{C}}}^{1,1}, \mathcal{M}_{\widehat{\mathcal{C}}}^{2,1}$.

Theorem 1.8. *Let $\widetilde{\mathcal{C}}$ and $\widehat{\mathcal{C}}$ be generic arrangements of cardinality m.*

1. $\mathrm{EDD}(\mathcal{M}_{\widetilde{\mathcal{C}}}^{1,1}) = 3m - 2$.
2. *If $m \geq 3$, then* $\mathrm{EDD}(\mathcal{M}_{\widehat{\mathcal{C}}}^{2,1}) = \frac{9}{2}m^2 - \frac{19}{2}m + 3$.

2. Numerical Experiments

In this section we conduct numerical experiments of the triangulation process of incident point and lines. All the code can be found in the Supplementary Material.

By (L1) we refer to point-line arrangements in $\mathbb{P}^3$ of one line incident to p points, as depicted in Figure 1. Our goal is to reconstruct such arrangements, given point correspondences incident to a single line correspondence across m views. The following is a list of natural approaches for such triangulation given a camera arrangement $\mathcal{C}$. It is not necessarily a complete list.

(L1).0 Triangulate each point correspondence by fitting it to the point multiview variety $\mathcal{M}_{\mathcal{C}}$, i.e. find the closest point correspondence in $\mathcal{M}_{\mathcal{C}}$;

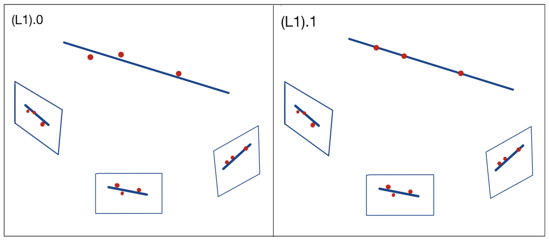

Figure 2: Comparison of (L1).0 and (L1).1 triangulation approaches. (L1).0 expects non-collinear points in the reconstruction, while (L1).1 produces collinear points.

(L1).1 Reconstruct the 3D line L by back-projecting the image lines from two views. Triangulate the point correspondences by fitting them to the anchored multiview variety $\mathcal{M}_{\mathcal{C}}^{L}$;

(L1).2 Triangulate two point correspondences by fitting them to $\mathcal{M}_{\mathcal{C}}$ to get 3D points X and Y. Let L be the line they span in $\mathbb{R}^3$. Triangulate the remaining point correspondences by fitting them to $\mathcal{M}_{\mathcal{C}}^{L}$;

(L1).3 Triangulate one point correspondence by fitting it to $\mathcal{M}_{\mathcal{C}}$ to get a 3D point X. Reconstruct the 3D line L by fitting the line correspondence to the anchored variety $\mathcal{L}_{\mathcal{C}}^{X}$. Triangulate the remaining point correspondences by fitting them to $\mathcal{M}_{\mathcal{C}}^{L}$;

(L1).4 Reconstruct the 3D line L by fitting the line correspondence to the line multiview variety $\mathcal{L}_{\mathcal{C}}$. Triangulate the point correspondences by fitting them to the anchored multiview variety $\mathcal{M}_{\mathcal{C}}^{L}$.

Approaches (L1).1-4 take the incidence relations into account in the triangulation, whereas (L1).0 does not. Therefore, in contrast to (L1).1-4, the resulting triangulation of (L1).0 does not preserve the point-line incidences; see Figure 2.

2.1. Implementation

Our experiments are implemented in `HomotopyContinuation.jl` [8] in `Julia` [5]. Here we explain them in some detail. The full explanation can be found in Supplementary Material. We ran the code on a Intel(R) Core(TM) i5-8300H CPU running at 2.30GHz.

2.1.1 Reducing the number of parameters

We reduce the number of parameters involved in the problem of fitting data to $\mathcal{M}_{\mathcal{C}}^{L}$, respectively $\mathcal{L}_{\mathcal{C}}^{X}$, by translating it to and solving it for $\mathcal{M}_{\widetilde{\mathcal{C}}}^{1,1}$, respectively $\mathcal{M}_{\widehat{\mathcal{C}}}^{2,1}$. Details are found in Supplementary Material Section B. This translation corresponds to a reduction of parameters because an affine patch of $\mathcal{M}^{1,1}_{\widetilde{\mathcal{C}}}$ lives in $(\mathbb{R}^1)^m$, while an affine patch of $\mathcal{M}^{L}_{\mathcal{C}}$ lives in $(\mathbb{R}^2)^m$. The analogous is true for $\mathcal{M}^{2,1}_{\widehat{\mathcal{C}}}$ and $\mathcal{L}^{X}_{\mathcal{C}}$.

We call the methods that do *not* use this translation *standard* and for instance, write (L1).1 std. to denote it. By just writing (L1).1, we denote the implementation of this translation.

2.1.2 Evaluating error

In our experiments, each iteration starts by randomly generating a 3D line L and p points X_i on this line. This line and these points are projected by randomly generated cameras to line and point correspondences in $(\mathbb{R}^2)^m$. For fixed $\epsilon = 10^{-12}$, randomly generated noise vectors $\sigma(\epsilon)$ of length ϵ are added in each factor. Our approaches then find points $Y_i \in \mathbb{R}^3$ that match the noisy correspondences. We measure the accuracy by taking the logarithm after averaging the relative error of reconstructed points:

$$e = \log_{10}\left(\frac{\sum \|Y_i - X_i\|}{p\epsilon}\right), \tag{13}$$

where $\|Y_i - X_i\|$ denotes the Euclidean distance. The interpretation of this number is that the error in the data gets amplified by 10^e during the triangulation.

2.1.3 Algorithms

The pseudocodes for all approaches are included in the Supplementary Material. The pseudocode for approach (L1).1 is presented in Algorithm 1. We use the notation that for a column vector $X \in \mathbb{R}^n$, $[X;\ 1] \in \mathbb{R}^{n+1}$ is the vector we get by adding a 1 as the last coordinate. Let ι be the function that scales a vector such that its last coordinate is 1, and then removes that coordinate. In lines 1 and 2 of the algorithm, we generate noisy data by introducing randomly generated noise $\sigma(\epsilon)$ as explained above, where $\epsilon = 10^{-12}$ is a fixed parameter for our experiments. Lines 3 and 4 correspond to the triangulation of the line and the point correspondences using the anchored point multiview variety at the line L_0. Note that we solve the closest point problem in line 4 by computing the zeros of a system of polynomial (critical) equations using the *solve* function in `HomotopyContinuation.jl` [9]. Finally, line 5 compares the points obtained in the previous step with the original starting points, measuring the logarithmic average relative error of the p points.

Most other pseudocodes are similar to Algorithm 1. For instance, approach (L1).0 does not take into account the line, so it corresponds deleting lines 1 and 3 from Algorithm 1 and modifying the minimization domain in line 4, replacing L_0 by $\mathcal{M}_{\mathcal{C}}$. In approach (L1).2, line 3 is modified so that L_0 is obtained by the span of two triangulated points instead of by intersecting two back-projected planes.

Algorithm 1: One iteration of the (L1).1 std. method given a randomly generated camera arrangement $\mathcal{C}$ of 3×4 matrices, a projective line L spanned by two vectors of $\mathbb{R}^4$, and p points $X_i \in \mathbb{R}^3$ such that $[X_i;\ 1]$ lie on L.

```
  Input  : $\mathcal{C} = (C_1, \dots, C_m)$, $L$, $X_1, \dots, X_p$
  Output: The log of the average relative error
1 for $j$ from 1 to $m$ do
2     for $i$ from 1 to $p$ do
3         $q_{i,j} \leftarrow \iota(C_j[X_i;\ 1]) + \sigma(\epsilon)$;
4     $u_j \leftarrow \iota(C_j \cdot L) + \sigma(\epsilon)$;
5 $L_0 \leftarrow \text{nullspace}\left[C_1^T[u_1;\ 1] \quad C_2^T[u_2;\ 1]\right]^T$;
6 for $i$ from 1 to $p$ do
7     $Y_i \leftarrow \underset{X\in\mathbb{R}^3:[X;\ 1]\in L_0}{\text{argmin}} \sum_{j=1}^m (q_{i,j} - \iota(C_j[X;\ 1]))^2$;
8 $e \leftarrow \log_{10}\left(\frac{1}{p\epsilon}\sum_{i=1}^p \|Y_i - X_i\|\right)$;
  Return: $e$
```

2.2. Numerical results

The main results of our numerical experiments are presented in Figures 3 to 5 and Tables 1 to 3. We compare the performance of the five different triangulation approaches for $p = 5$ and different number of m cameras. In the tables, we present the median, mean and standard deviation σ of the logarithmic average relative error and time. The results are displayed in histograms and tables created with `Plots.jl` [10].

Note that for two cameras, (L1).0 and (L1).2 are the exact same, and (L1).1 and (L1).4 are essentially the same. In Figure 3 and Table 1 we present results for $m = 2$ and therefore only include (L1).0, (L1).1 and (L1).3. We also compare with the standard implementations of (L1).1 and (L1).3 that do not use the linear isomorphisms of Theorem 1.3 to improve computation time as described in Section 2.1.1. This simulation is iterated 1000 times.

Figure 4 and Table 2 compare the different reconstruction approaches for $m = 3$ cameras. The simulations are again iterated 1000 times.

Figure 5 and Table 3 compare the different reconstruction approaches for $m = 4$ cameras. The simulations are iterated only 100 times, due to the longer run-time of the simulations.

Our theoretical studies in Section 1, specifically Theorem 1.7, guarantee that methods such as (L1).1, (L1).2 and (L1).3 are less complex from the algebraic point of view, than (L1).0 for triangulation of points incident to a line. Ad-

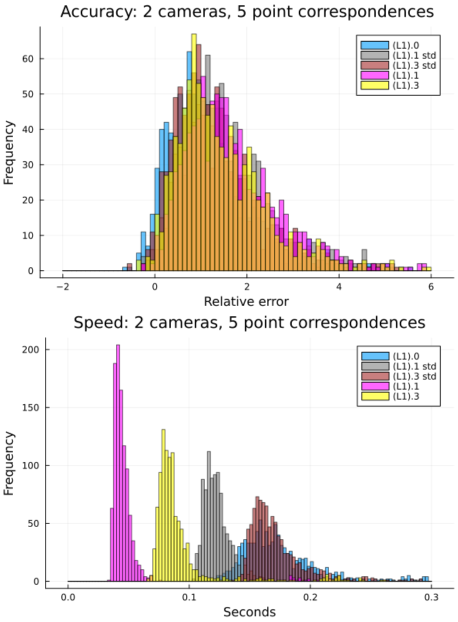

Figure 3: Triangulation of $p = 5$ point correspondences incident to a line for $m = 2$ views with complete visibility. 1000 iterations. The histograms show the frequency of the average relative error and the running time of the depicted triangulation methods.

ditionally, using Section 2.1.1, we were able to improve the computation speed while retaining comparable accuracy, which is exemplified by the two cases (L1).1 vs. (L1).1 std. and (L1).3 vs. (L1).3 std. in Table 1.

Finally, our numerical simulations show that in our `HomotopyContinuation.jl` implementation, the (L1).1 approach is the fastest across all number of views, but least accurate. In the case of $m = 3$, (L1).4 is the most accurate and faster than the traditional (L1).0 method. Even for $m = 4$, (L1).4 is the most accurate but is notably slower than (L1).0. Increasing the number of cameras m improves accuracy and reduces speed, but the exact impact depends on the approach. Among (L1).1-4, (L1).1 is least affected in both regards, and (L1).4 is most affected in both regards. The speed can be thought of as an affine function in the number of point correspondences p. Reconstructing the line correspondence takes some fixed amount of time independent of p and then all p correspondences are independently reconstructed.

3. Conclusion and Future Work

This work studied the triangulation of incident points and lines. We introduced the anchored line multiview varieties and showed that they correspond to multiview varieties arising from projections $\mathbb{P}^2 \dashrightarrow \mathbb{P}^1$ and $\mathbb{P}^1 \dashrightarrow \mathbb{P}^1$. In addition, we proved that they are less complex for reconstruction via critical points. These theoretical results are also aligned with the numerical experiments we conducted. In particular, the proposed methods in Section 2 compare different methods for triangulating a set of points incident to a line. We highlight that according to our experiments, the use of the theoretical results in Section 1 allows for a notably faster triangulation while preserving the accuracy of the traditional method for point reconstruction.

We hope this work motivates the use of new algebraic constraints in the triangulation of incident points and lines.

Accuracy	median	mean	σ
(L1).0	1.031	1.243	1.213
(L1).1	1.489	1.702	1.085
(L1).1 std.	1.374	1.548	0.990
(L1).3	1.250	1.449	0.964
(L1).3 std.	1.147	1.393	1.205

Speed	median	mean	σ
(L1).0	0.170	0.201	0.220
(L1).1	0.0430	0.0470	0.0207
(L1).1 std.	0.122	0.141	0.202
(L1).3	0.0838	0.0941	0.0464
(L1).3 std.	0.167	0.188	0.0764

Table 1: Triangulation of $p = 5$ point correspondences incident to a line for $m = 2$ views with complete visibility. 1000 iterations. The tables show the accuracy and speed of the depicted triangulation methods in terms of median, mean and standard deviation.

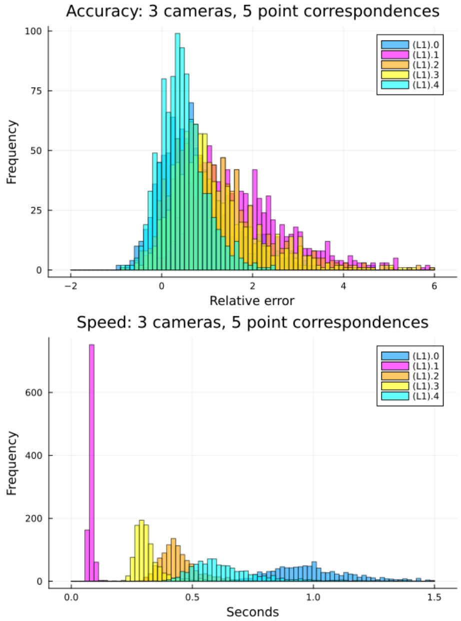

Figure 4: Triangulation of $p = 5$ point correspondences incident to a line for $m = 3$ views with complete visibility. 1000 iterations. The histograms show the frequency of the average relative error and the running time of the depicted triangulation methods.

Accuracy	median	mean	σ
(L1).0	0.620	0.817	1.023
(L1).1	1.562	1.781	1.128
(L1).2	0.878	1.125	0.998
(L1).3	1.011	1.243	1.078
(L1).4	0.407	0.499	0.901

Speed	median	mean	σ
(L1).0	0.984	1.060	0.514
(L1).1	0.0806	0.0858	0.0318
(L1).2	0.431	0.456	0.155
(L1).3	0.304	0.331	0.134
(L1).4	0.616	0.763	0.500

Table 2: Triangulation of $p = 5$ point correspondences incident to a line for $m = 3$ views with complete visibility. 1000 iterations. The tables show the accuracy and speed of the depicted triangulation methods in terms of median, mean and standard deviation.

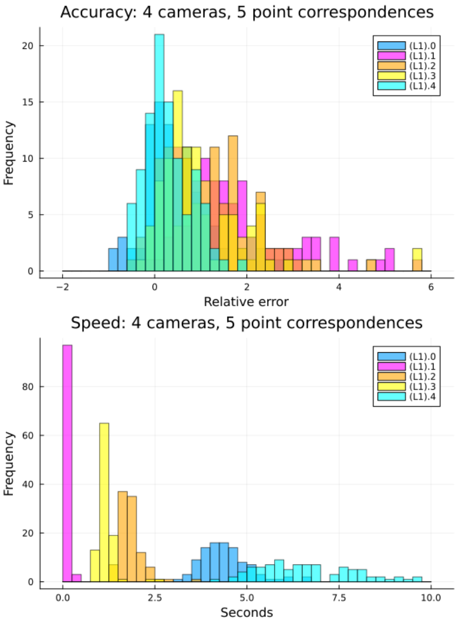

Figure 5: Triangulation of $p = 5$ point correspondences incident to a line for $m = 4$ views with complete visibility. 100 iterations. The histograms show the frequency of the average relative error and the running time of the depicted triangulation methods.

Some questions that arose in the development of this work, and that we believe are worth studying, are included below.

- A triangulation approach inspired by [2] consists of reconstructing the 3D line L by fitting it to $\mathcal{L}_\mathcal{C}$ solving the following optimization problem

$$\text{minimize} \quad \sum_{ij} d(x_{ij}, \ell_i)^2 \text{subject to} \quad \ell \in \mathcal{L}_\mathcal{C}. \quad (14)$$

Accuracy	median	mean	σ
(L1).0	0.385	0.551	0.776
(L1).1	1.437	1.762	1.249
(L1).2	1.095	1.077	0.845
(L1).3	0.817	1.110	1.094
(L1).4	0.203	0.421	1.324

Speed	median	mean	σ
(L1).0	4.365	4.675	1.576
(L1).1	0.141	0.149	0.0431
(L1).2	1.809	1.863	0.323
(L1).3	1.107	1.157	0.234
(L1).4	6.599	7.391	3.068

Table 3: Triangulation of $p = 5$ point correspondences incident to a line for $m = 4$ views with complete visibility. 100 iterations. The tables show the accuracy and speed of the depicted triangulation methods in terms of median, mean and standard deviation.

Here, x_{ij} denotes the $j = 1, \ldots, p$ point correspondences across $i = 1, \ldots, m$ cameras, and $d(x_{ij}, \ell_i)$ is the affine distance from a line in $\mathbb{R}^2$ to the point, meaning

$$d(x_{ij}, \ell_i) = \frac{|1 + (\ell_i)_1 (x_{ij})_1 + (\ell_i)_2 (x_{ij})_2|}{\sqrt{(\ell_i)_1^2 + (\ell_i)_2^2}}. \quad (15)$$

We believe that this method is more accurate than (L1).4, but with longer running time, due to the appearance of fractions.

- Consider the point-line problem (P1) defined by one point incident to l lines, illustrated in Figure 1. Can similar approaches to (L1).1-4 be formulated for the triangulation of the (P1) setting? Can they be used to improve accuracy and/or the speed of triangulation?

- Our implementation for the numerical experiments does not directly correspond to the specialized software and hardware used for triangulation in practice. However, our results motivate that patterns displayed in our figures and tables could translate to such settings. This could improve the efficiency of current specialized solvers.

Acknowledgements. The authors thank Kathlén Kohn for helpful discussions, Lukas Gustafsson for the initial insight of Theorem 1.3, and Viktor Larsson for pointing out relevant references at the start of this project. Felix Rydell and Angélica Torres were supported by the Knut and Alice Wallenberg Foundation within their WASP (Wallenberg AI, Autonomous Systems and Software Program) AI/Math initiative. Elima Shehu is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Projektnummer 445466444. Angélica Torres is currently supported by the Spanish State Research Agency, through the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D.

References

[1] Sameer Agarwal, Andrew Pryhuber, and Rekha R Thomas. Ideals of the multiview variety. IEEE transactions on pattern analysis and machine intelligence, 2019. 1, 2

[2] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. Computer Vision and Image Understanding, 100(3):416–441, 2005. 3, 8

[3] Paul A Beardsley, Andrew Zisserman, and David William Murray. Navigation using affine structure from motion. Lecture Notes in Computer Science, 801:85–96, 1994. 2, 3

[4] Paul A Beardsley, Andrew Zisserman, and David William Murray. Sequential updating of projective and affine structure from motion. International journal of computer vision, 23:235–259, 1997. 2, 3

[5] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. arXiv preprint arXiv:1209.5145, 2012. 5

[6] Paul Breiding, Timothy Duff, Lukas Gustafsson, Felix Rydell, and Elima Shehu. Line multiview ideals. arXiv preprint arXiv:2303.02066, 2023. 2

[7] Paul Breiding, Felix Rydell, Elima Shehu, and Angélica Torres. Line multiview varieties. SIAM Journal on Applied Algebra and Geometry, 7(2):470–504, 2023. 1, 2, 4

[8] Paul Breiding and Sascha Timme. Homotopycontinuation. jl: A package for homotopy continuation in julia. In International Congress on Mathematical Software, pages 458–465. Springer, 2018. 5

[9] Paul Breiding and Sascha Timme. HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In Mathematical Software – ICMS 2018, pages 458–465, Cham, 2018. Springer International Publishing. 2, 6

[10] Tom Breloff and other contributors. JuliaPlots/Plots.jl. 6

[11] Jan Draisma, Emil Horobeţ, Giorgio Ottaviani, Bernd Sturmfels, and Rekha R Thomas. The euclidean distance degree of an algebraic variety. Foundations of computational mathematics, 16(1):99–149, 2016. 2, 4

[12] Timothy Duff, Kathlen Kohn, Anton Leykin, and Tomas Pajdla. Plmp-point-line minimal problems in complete multi-view visibility. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1675–1684, 2019. 1

[13] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomas Pajdla. Pl$$_1$$p - point-line minimal problems under partial visibility in three views. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, Computer Vision – ECCV 2020, pages 175–192, Cham, 2020. Springer International Publishing. 1

[14] A. Elqursh and A. Elgammal. Line-based relative pose estimation. In Conference on Computer Vision and Pattern Recognition, 2011. 3

[15] Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret H Regan, David da Costa de Pinho, Elias Tsigaridas, Charles W Wampler, Jonathan D Hauenstein, Peter J Giblin, Benjamin Kimia, et al. Trplp-trifocal relative pose from lines at points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12073–12083, 2020. 1

[16] Olivier Faugeras. Three-dimensional computer vision: A geometric viewpoint. MIT press, Cambridge, MA, 1993. 1

[17] Olivier Faugeras and Bernard Mourrain. On the geometry and algebra of the point and line correspondences between n images. In Proceedings of IEEE International Conference on Computer Vision, pages 951–956. IEEE, 1995. 1, 2

[18] Andreas Gathmann. Algebraic geometry, 2019/20. Class Notes TU Kaiserslautern. Available at https://www.mathematik.uni-kl.de/~gathmann/de/algeom.php. 4

[19] Corey Harris and Daniel Lowengrub. The chern-mather class of the multiview variety. Communications in Algebra, 46(6):2488–2499, 2018. 2

[20] Richard I. Hartley. Lines and points in three views and the trifocal tensor. International Journal of Computer Vision, 22(2):125–140, 1997. 2, 3

[21] Richard I Hartley and Peter Sturm. Triangulation. Computer vision and image understanding, 68(2):146–157, 1997. 2

[22] Richard I. Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 2

[23] Manuel Hofer, Andreas Wendel, and Horst Bischof. Incremental line-based 3d reconstruction using geometric constraints. The British Machine Vision Association and Society for Pattern Recognition, 2013. 3

[24] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 626–635, 2018. 1

[25] Javier Civera Joan Solà, Teresa Vidal-Calleja and José María Martínez Montiel. Impact of landmark parametrization on monocular ekf-slam with points and lines. International Journal of Computer Vision, 97(3):339–368, 2012. 3, 9

[26] Kenichi Kanatani. Statistical optimization for geometric computation: theory and practice. Courier Corporation, 2005. 2

[27] Kenichi Kanatani, Yasuyuki Sugaya, and Hirotaka Niitsuma. Triangulation from two views revisited: Hartley-sturm vs. optimal correction. practice, 4(5):99, 2008. 2

[28] Yasushi Kanazawa and Kenichi Kanatani. Reliability of 3-d reconstruction by stereo vision. IEICE TRANSACTIONS on Information and Systems, 78(10):1301–1306, 1995. 2

[29] Joseph David Kileel. Algebraic Geometry for Computer Vision. ProQuest LLC, Ann Arbor, MI, 2017. Thesis (Ph.D.)–University of California, Berkeley. 1, 2

[30] Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. 3d line mapping revisited. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 21445–21455, June 2023. 1

[31] Laurentiu G. Maxim, Jose I. Rodriguez, and Botong Wang. Euclidean distance degree of the multiview variety. SIAM Journal on Applied Algebra and Geometry, 4(1):28–48, 2020. 1, 2

[32] Branislav Micusik and Horst Wildenauer. Structure from motion with line segments under relaxed endpoint constraints. In 2014 2nd International Conference on 3D Vision, volume 1, pages 13–19, 2014. 3

[33] Long Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(8):834–845, 1997. 3

[34] Grant Schindler, Panchapagesan Krishnamurthy, and Frank Dellaert. Line-based structure from motion for urban environments. Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), pages 846–853, 2006. 1

[35] Henrik Stewénius, Frederik Schaffalitzky, and David Nistér. How hard is 3-view triangulation really? In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, volume 1, pages 686–693. IEEE, 2005. 3

[36] Matthew Trager, Martial Hebert, and Jean Ponce. The joint image handbook. In Proceedings of the IEEE international conference on computer vision, pages 909–917, 2015. 2, 4