Learning to Solve Hard Minimal Problems*

Petr Hruby
ETH Zürich
Department of Computer Science
petr.hruby@inf.ethz.ch

Timothy Duff
University of Washington
Department of Mathematics
timduff@uw.edu

Anton Leykin
Georgia Institute of Technology
School of Mathematics
leykin@math.gatech.edu

Tomas Pajdla
Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
pajdla@cvut.cz

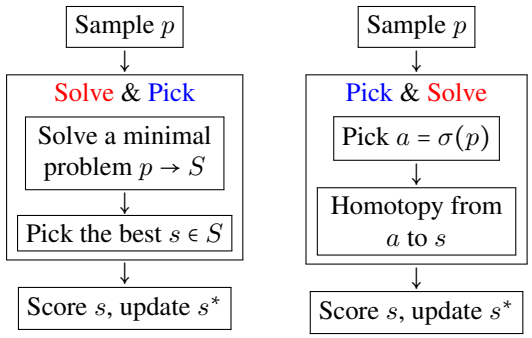
Abstract

We present an approach to solving hard geometric optimization problems in the RANSAC framework. The hard minimal problems arise from relaxing the original geometric optimization problem into a minimal problem with many spurious solutions. Our approach avoids computing large numbers of spurious solutions. We design a learning strategy for selecting a starting problem-solution pair that can be numerically continued to the problem and the solution of interest. We demonstrate our approach by developing a RANSAC solver for the problem of computing the relative pose of three calibrated cameras, via a minimal relaxation using four points in each view. On average, we can solve a single problem in under 70 μs. We also benchmark and study our engineering choices on the very familiar problem of computing the relative pose of two calibrated cameras, via the minimal case of five points in two views.

1. Introduction

Minimal problems arise from geometrical problems in 3D reconstruction [59, 61, 62], image matching [55], visual oodometry,and localization [3, 49, 57, 66]. Many geometrical problems have been successfully formulated and solved as minimal problems [1, 4–6, 11, 12, 18, 27, 34–37, 45, 46, 48, 54, 56, 58, 64, 67]. Technically, minimal problems are systems of polynomial equations which depend on the input data and have a finite number of solutions.

*This work was partially supported by projects: EU RDF IMPACT No. CZ.02.1.01/0.0/0.0/15 003/0000468, EU H2020 ARtwin No. 856994. The research of AL is partially supported by NSF DMS-2001267. TD acknowledges support from an NSF Mathematical Sciences Postdoctoral Research Fellowship (DMS-2103310). We thank Dmytro Mishkin, Viktor Larsson, and Zuzana Kukelova for their thoughtful comments.



(a) The standard use of minimal problems in RANSAC calls for solving and scoring a large number of spurious solutions in S.

(b) We suggest to learn a picking function σ that finds start parameters a from which the homotopy continuation reaches s.

Figure 1. The inner RANSAC loop finds the best solution for a data sample p. This is very expensive when a minimal problem has many spurious solutions. Our efficient homotopy continuation combined with machine learning avoids solving for the spurious solutions. Thus, using minimal problems in RANSAC becomes effectively independent from the number of spurious solutions.

1.1. Motivation

Many geometrical problems are optimization problems that have only one optimal solution. Minimal problems, however, often have many additional spurious solutions. The optimal solution is typically real, satisfies inequality constraints, and fits well all data. Such constraints, however, can not be used by methods of nonlinear algebra [13, 65] which have no ability to bypass finding (or incurring the cost of finding) all solutions of polynomial systems.

RANSAC [23, 53] approximates the optimal solution to a geometrical problem by computing candidate solutions from data samples and picking a solution with maximal data support. This is done by iterating over the samples in

an outer loop and over the solutions of a minimal problem for each sample in an inner loop. To find a single solution for a data sample in the inner loop, the state-of-the-art "solve & pick" approach first computes all solutions of a minimal problem and then picks the optimal solutions by removing nonreal solutions, using inequalities, and evaluating the support. Optimization in the inner loop may be very costly when there are many spurious solutions to the minimal problem. Fig. 1 compares the standard "solve & pick" approach with our "pick & solve" approach that learns, for a given data sample, how to first pick a promising starting point and then (ideally) continue it to a meaningful solution.

Recent results [15, 16] show that there are many minimal problems in multiview geometry with many spurious solutions which the state-of-the-art polynomial solvers cannot solve efficiently. [1]

1.2. Contribution

We present a method for combining optimized homotopy continuation (HC) with machine learning to avoid solving for spurious solutions. The main idea is to learn a single starting point for a real HC path that has a good chance to reach a good solution of the original geometrical problem.

To demonstrate our method on a hard problem, we develop an efficient solver for the "Scranton" minimal problem obtained by relaxing the overconstrained problem of four points in three views (4pt) [50]. We train a model that predicts a starting problem for a single path real HC method to find a good solution. Our solver is implemented efficiently in C++ and evaluated on the state-of-the-art data in computer vision. It successfully solves about 26.3% of inputs in 16.3μs, Tab. 4. In Sec. 9 we show that when used in RANSAC, about 4 samples suffice on average to obtain a valid candidate of camera geometry in 61.6μs. No such efficient solver has been known for this problem before. The best-known runtime for a very carefully designed approximation of the problem, reported in [50], was on the order of milliseconds. We thus achieve more than ten times speedup compared to [50]. Most importantly, our approach is general and opens the door to solving other hard minimal problems, e.g., from [15, 16].

We benchmark (Sec. 9) our approach on the classical 5-point problem (5pt) [48] using standard benchmarks [31]. We show that for the 5pt problem, we can solve 29.0% of inputs in about 7.6μs, Tab. 4. Thus, in RANSAC, we can solve it in average in 26.1μs.

Our approach is general. It can be applied even in some cases where the number of spurious solutions is not finite. For instance, our depth formulation of the Scranton problem has an infinite family of solutions where some depths may be zero. Additional polynomial constraints, which do not need to be explicitly enforced, reduce the number of potential solutions to 272—see 15. Thus, by exploiting the "locality" of HC methods, we can guarantee that when starting from a good starting point, we can ignore other spurious solutions with no additional computational cost.

It is important to highlight that, unlike the current symbolic-numeric solvers, our method is coupled with the rest of SfM pipeline, i.e., it uses the real data distribution in a particular vision problem at hand.

1.3. Previous work

The state-of-the-art approach to solving polynomial systems in computer vision is based on symbolic-numeric solvers, which combine elimination (by Gröbner bases [37, 40, 64] or resultants [8, 19, 29]) with eigenvector computation [65] to find all complex solutions. Currently, symbolic-numeric solvers [37, 40–42] provide efficient and stable results for minimal problems with as many as 64 complex solutions [8, 43]. However, these solvers mostly fail to deliver practical results for hard computer vision problems with many solutions. Symbolic/numeric solvers involve two hard computational tasks. First, in the symbolic part, large matrices are constructed and triangularized by the Gauss-Jordan elimination. Secondly, in the numeric part, the eigenvectors of $n \times n$ matrices, where n is the number of all complex solutions, are computed. Both steps are prohibitive for generic polynomial systems with many spurious solutions. Methods which deal with real solutions only, e.g. Sturm sequences [48], also generally require expensive manipulations (reduction to a univariate polynomial) and may still need to consider many spurious real solutions.

Global HC methods give an alternative, well-studied approach [7, 10, 14, 68] to finding *all complex solutions* of polynomial systems. Off-the-shelf HC solvers [7, 10, 14, 68] have been proven useful for studying the structure of minimal problems [2, 30, 32, 52]. However, the off-the-shelf solvers are much slower (10^3 – 10^5 times) than current symbolic-numeric solvers. For instance, our experiments, Tab. 7, show that solving [48] with complex homotopy continuation in Macaulay2 [44] takes about $10^5 \mu s$ compared to $5\,\mu s$ when [39] implementation of [48] is used.

The previous work [20] closest to this work addresses the problem of speeding up minimal HC solvers. This paper took notable steps towards the practical use of homotopy continuation for minimal problem solving. In that work, an off-the-shelf HC implementation [44] has been optimized and efficiently implemented in modern C++. Two hard problems in trifocal geometry have been solved in about 660ms. Despite not providing practical solvers for RANSAC [23, 53], the results of [20] demonstrated that hard multiview problems involving 312 and 216 solutions can be solved in a stable way and thus could be useful for building practical structure from motion algorithms.

Our paper considers a novel solver for a problem named

[1] See Sec. 12 in the SM for more about these problems.

"Scranton"[2], which is a minimal relaxation of the overconstrained 4pt problem [50]. Previous work formulated Scranton using camera parameters and found that it has 272 complex solutions [16, 32]. We consider an alternative formulation in terms of 3D point depths, analogous to [52], which has 272 potentially meaningful solutions. The original 4pt problem was solved by numerical search in [50], with about $1ms$ runtime. A depth-formulated 4pt problem was also studied in [52], showing that the overconstrained problem with exact input has a unique solution. Their exact solution does not apply to problems with noisy data.

We also study the classical, well-understood 5-point problem (5pt) of computing the relative pose of two calibrated cameras [48]. Unlike Scranton, this problem has many practical solutions [9, 27, 38, 48, 59]. Currently, the most efficient symbolic-numeric solver [42] of the 5pt problem solves for up to 10 essential matrices in about $5\mu s$. Although the 5pt problem is not as hard as Scranton in terms of the number of spurious solutions, it does provide an important testing ground for us.

2. Our approach

Here we present our approach to solving hard minimal problems. We shall use HC methods to track one real solution of a start problem to obtain one real solution of the target problem. We shall design an algorithm such that this one solution we obtain is a *meaningful* solution with sufficient success rate.

2.1. Problem-solution manifold

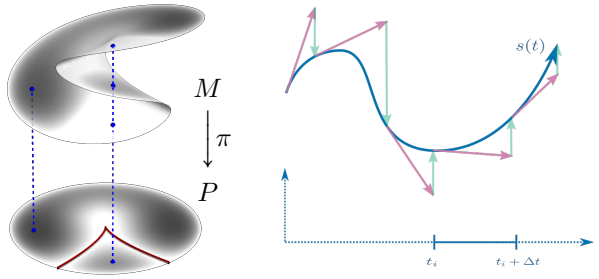
We operate in the *problem-solution manifold* M of *problem-solution (p-s) pairs* (p, s), where p is a problem and s is a solution of p. Problem p belongs to a *real* vector space P. Solution s comes from a *real* vector space of solutions. The projection $\pi : M \to P$ is defined by $(p, s) \mapsto p$. The preimage $\pi^{-1}(p) \in M$ contains all p-s pairs that correspond to a particular problem p.

Example 1. *To illustrate the introduced concepts, let us look at one equation $x^3 + ax + b = 0$ in one unknown x with two parameters a and b. Here a problem $p = (a, b)$ has either one or three real solutions depending on whether the discriminant $D = 4a^3 + 27b^2$ is positive or negative; see the corresponding problem-solution manifold M in Fig. 2a.*

To be precise, the equation defines an algebraic variety, which is guaranteed to be a smooth manifold when points above the discriminant locus $D = 0$ are removed.

See SM Sec. 13 for the detailed examples of setting up problem-solution manifolds for the 5pt problem and Scranton, a minimal relaxation for the 4pt problem. Below is a condensed version of the 5pt problem.

[2]The US telephone area code for Scranton PA is 272.



(a) (b)

Figure 2. (a) Problem-solution manifold M projected to the problem space P. (b) Numerical HC method.

Example 2. *Consider the classical 5pt problem of computing the relative pose of two calibrated cameras which view five world points where the scale is fixed such that the first 3D point lies in the first image plane.*

Denote by x the images of 5 points in 2 views: i.e., x is a point in $P = \mathbb{R}^{20}$. Assume the points are in front of both cameras: i.e., their depths $\lambda_{i,j}, (i = 1, \dots, 5; j = 1, 2)$ are all positive. Our assumptions imply that $\lambda_{1,1} = 1$. The unknown depths vector in the solution space $\mathbb{R}^9$ determines the relative pose. Therefore, the problem-solution manifold M is contained in the ambient space $\mathbb{R}^{20} \times \mathbb{R}^9$. Equations vanishing on M are given in Sec. 7.1.

2.2. Probability distribution on M

We introduce a probability density μ on the problem-solution manifold M that gives the distribution of real-world problem-solution pairs. In Fig. 2a, we give an example of μ depicted with shades of gray (darker is bigger) on the manifold M. Note that the density is such that any problem with a set S of three solutions has only one $s \in S$ with $\mu(s) > 0$. We shall (implicitly) operate under the following assumption:

An input problem p is likely to have one meaningful solution that is dominant, i.e., occurs much more frequently in the real data than other meaningful solutions.

In many problems (e.g., the one in Section 7.2) the number of meaningful solutions is guaranteed to be exactly one generically. The distribution μ is hard to model; in what follows it is represented by training data.

2.3. Pick & solve vs. solve & pick

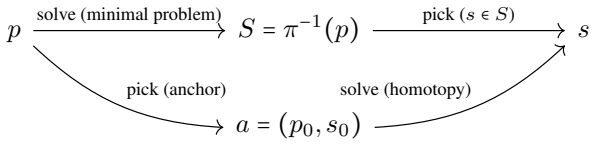
A typical local iterative method (e.g. Newton's method, gradient descent, etc.) would attempt solving a problem p by obtaining an initial approximate solution s_0 with a hope that it is not too far away from an actual solution and then producing a sequence of its refinements $s_0, s_1, s_2, \dots$ until either a desired quality is reached or some termination-with-failure criterion is satisfied.

Our homotopy approach is a generalization of such local methods. In a nutshell, given a problem p,

1. we select a suitable start problem-solution pair $(p_0, s_0) \in M$ for p,
2. we choose a path $p_0 \rightsquigarrow p$ in the problem space P, leading from p_0 to p,
3. we track the path $(p_0, s_0) \rightsquigarrow (p, s)$ to obtain the target solution s of p.

Selecting a start pair (p_0, s_0) is the key ingredient of our approach. Given a real HC method, one can aim to construct the selection strategy $\sigma(p) = (p_0, s_0)$ in two steps. First, one finds a small set of *anchors* $A \subset M$, such that it is possible to reach (cover) a significant part of M from A by the real homotopy continuation. Secondly, one learns a selection strategy σ such that starting from $(p_0, s_0) \in A$, the meaningful problem-solution pair is reached with sufficiently high frequency to make RANSAC work.

Intuitively, one may perceive a minimal solver employed in RANSAC as an arrow $p \to S$ in the following diagram:



where an instance p of a minimal problem is "solved" (all solutions in S are found.) Then RANSAC "picks" at most one solution, $S \to s$, a candidate that maximizes the number of inliers.

Looking for a shortcut that would allow us to go directly $p \to s$, we reverse this flow: first "picking" an anchor and then tracking *one* HC path to "solve".

2.4. Structure of our solvers

Our solvers for both 5pt and Scranton minimal problems have a common structure, consisting of an offline training stage and an online evaluation stage. Offline computations may be resource-intensive; however, the online stage must be very efficient to achieve practical sub-millisecond run times. The offline stage consists of:

1. Sampling data D, according to μ, representing the (preprocessed) problem-solution manifold M (Sec 3).
2. Covering a sufficient fraction of the data with anchors $A \subset D$ (Sec. 4).
3. Learning a model σ which selects a starting p-s pair $(p_0, s_0) \in A$ for any given problem p (Sec. 5).

The online stage consists of:

1. Preprocessing the input p to reduce its variability (Sec. 8).
2. Selecting a starting pair from A as $(p_0, s_0) = \sigma(p)$.
3. Constructing polynomial equations from p (Sec. 7).
4. Computing a solution s of p by HC from (p_0, s_0) (Sec. 6).

The next sections describe these steps in detail.

	5pt problem				Scranton			
n	1K	4K	10K	40K	1K	4K	10K	40K
50 %	8	9	8	8	18	18	17	16
75 %	25	28	27	26	47	51	50	50
90 %	59	70	70	70	92	112	120	134
95 %	*90*	109	115	124	*126*	168	191	233
100 %	*140*	*235*	*334*	*585*	*176*	*335*	*507*	*1205*

Table 1. Number of anchors from *office* + *terrains* (generated as described in Sec. 4, cf. Tab. 2) which are needed to cover 50%, 75%, 90%, 95%, and 100% of n problem-solution pairs. Different values of n are considered. A problem-solution pair is covered by the set of anchors if there is at least one anchor from which the problem-solution pair can be correctly tracked. A track is considered correct if the Euclidean distance from the obtained solution to the ground-truth solution is less than 10^{-5}.

3. Sampling data representing M and μ

The offline stages of our solvers begin by sampling the data D given by 3D models of various realistic objects. We use models from ETH 3D Dataset to represent μ.

A 3D model consists of 3D points X, cameras C, and relation $I \subset X \times C$ encoding observations ($(X_m, C_i) \in I$ iff C_i observes X_m). For Scranton, we may sample a single p-s pair $(p, s) \in M$ as follows:

1. Select 3 cameras $C_i, C_j, C_k \in C$
2. Select 4 points $X_l, X_m, X_n, X_o \in X$: $(X_a, C_b) \in I \; \forall a \in \{l, m, n, o\}, b \in \{i, j, k\}$
3. Project the points to the cameras to get 12 2D points $x_{a,b}$, concatenate them to 24-dim vector $p \in P$
4. Get the depths $\lambda_{a,b}$ of the points in the cameras, concatenate them to 12-dim vector $s \in S$

Sampling for the 5pt problem is similar; in step 1 we select 2 cameras, and in step 2 we select 5 points.

4. Selecting anchors A

We now describe how the starting p-s pairs are obtained. Our goal is to find a small set A of starting p-s pairs (the set of anchors), from which a high portion of p-s pairs from a given distribution can be tracked by HC.

If we limit ourselves to a finite set of p-s pairs and the anchors are selected from the same set, the optimal anchor selection procedure consists of building a graph with p-s pairs as vertices. Two vertices (p_i, s_i), (p_j, s_j) in this graph

Source	α [%]	Source	α [%]
Courtyard	78.1	Relief 2	77.0
Office	81.1	Off. + Terr.	**82.2**
Terrains	79.0	Off. + Rel.	80.7
Playground	75.4	O + T + P + R2	79.7

Table 2. Study of sources for anchor selection. Rows correspond to different models or combinations of models, from which the anchors are generated. For each source of anchors, we measure the percentage α of testing p-s problems (generated from models *delivery_area*, *electro*, *facade*, *kicker*, *meadow*, *pipes*) that can be reached from any of the anchors generated from the given source. A set of 100 anchors is considered for each source.

	5 pt problem coverage [%]				
# anchors	8	26	70	124	585
delivery_area	43.3	73.5	86.8	91.4	96.5
facade	51.3	74.9	88.6	92.9	97.0
	4 pt problem coverage [%]				
# anchors	16	50	134	233	1205
delivery_area	47.4	73.6	88.2	92.9	96.6
facade	42.1	71.1	87.9	92.9	96.9

Table 3. Percentage of testing problem-solution pairs covered by anchors generated from *office* and *terrains*. The anchors used are a subset of the n = 40000 used in Tab. 1. A p-s pair is "covered" if HC starting from some anchor yields a solution whose Euclidean distance from the ground-truth solution is less than 10^{-5}. This is an "oracle strategy" that always starts from the best anchor.

are connected with an edge if the correct solution s_j can be obtained by tracking HC from (p_i, s_i) to p_j. A set of anchors covering all problems in this graph is called a *dominating set*. Since computing a minimum-size dominating set is NP-hard, we use a greedy proxy which is known to perform well[3].

In Tab. 1, we generate n = 1K, 4K, 10K, 40K p-s pairs from the models *office* and *terrains*[4] for both the 5pt and Scranton problems. In each case, we use the greedy dominating set heurstic to determine a set of anchors which cover a given percentage of all generated p-s pairs. The results show that if the number of vertices in the graph is sufficiently high, a reasonable portion of different p-s pairs from the same scene can be solved by HC starting from one of the anchors obtained by our greedy strategy.

The choice of these models used in Tab. 1 is motivated by another experiment measuring the percentage of test problems covered by different sources of anchor selection. This experiment is summarized in Tab. 2, which shows that using the combination of *office* and *terrains* as a source of anchors generalizes best when compared to other sources.

Finally, Tab. 3 displays the rate at which HC starting from a fixed subset of these anchors successfully leads to the correct solution on test sets of 40K p-s pairs from *delivery_area* and *facade*. The resulting percentage may be interpreted as the success rate of an "oracle strategy" that always finds the best anchor to start from. The results give us a refined picture of how well the anchors we generate generalize to test scenes. Even when the number of anchors is small, eg. 16, the percentage of testing p-s pairs that are covered may be reasonably large, eg. 40–50%.

5. Learning σ to select the starting p-s pair

We formulate the problem of finding the best starting p-s pair as a classification task. Our method relies on a classifier σ, which for a sample problem $p \in P$ assigns a label from $\sigma(p) \in A \cup \{TRASH\}$, where A is the anchor set generated in Sec. 4. The label $TRASH$ is included for cases where no problem in A covers p.

Our goal is to minimize the *effective time* $\epsilon_t = \mu_t/\rho$ of the solver, where μ_t is the total time[5] and ρ is the success rate. Therefore, we must be able to classify p very fast, on the order of $10\mu s$ for a subsequent HC path $\sigma(p) \rightsquigarrow (p, s)$ $\sigma(p) \neq \{TRASH\}$.) Now, we are going to describe the classifier and its training.

For both problems, we use a Multi-Layer Perceptron (MLP) with 6 hidden layers of 100 neurons with bias.[6] The input layer has size $\dim P$, and the output layer has size $|A| + 1$. We use the PReLU activation function. During training, we use the dropout before the last layer to prevent overfitting. The classification time of the MLP is about $8\mu s$ for both 5pt and 4pt problems.

The input to the MLP is a normalized (Sec. 8) problem $p \in P$. The output is a vector of $|A| + 1$ numbers, which give the score for every starting p-s pair (p_0, s_0), as well as for $TRASH$. If the score of $TRASH$ is higher than the scores of all anchors, we skip the sample. Otherwise, we track from the p-s pair with the highest score.

During training, we normalize the output of the MLP with a softmax layer, and we use the standard cross-entropy loss. We use the SGD optimizer, which gives us better results than other optimizers, such as Adam. We generate training and testing data as described in Sec. 5.1, and train the MLP by minimizing the loss on the training data for 80 epochs. Then, we select the hyperparameters which maximize the success rate on the validation data. We evaluate the resulting classifier, Sec. 5.2.

[3]The proposed method is illustrated in SM. Fig. 5.
[4]https://www.eth3d.net/datasets
[5]preprocessing, anchor selection, tracking, and RANSAC scoring
[6]See SM Tab. 11 for a comparison with MLPs with different sizes.

5.1. Training data generation

We use the training and testing data generated from ETH 3D Dataset. Testing data is generated from models *delivery_area* and *facade*, training data from 23 other sequences. First, we generated p-s pairs (p, s) from the models according to Sec. 3. Then, we normalized each problem p (Sec. 8), and tracked the solution to problem p from each anchor $(\breve{p}_a, \breve{s}_a) \in A$. If the solution to p obtained by HC starting in anchor $(\breve{p}_a, \breve{s}_a) \in A$ is equal to the expected solution s, then the ID a of the anchor is assigned as the label of problem p. If solution s cannot be reached from any anchor, the label of p is $TRASH$. A problem may have multiple labels. We note that this procedure allows us, in principle, to generate an unlimited amount of training data[7].

Our experiments use $\approx$ 1M training p-s pairs per model (23M in total) and 30000 validation p-s pairs per model.

5.2. Classifier evaluation

In the evaluation of the trained MLPs, an anchor (p_0, s_0) is selected by the classifier, and HC is tracked from (p_0, s_0). **Success rate** is the percentage of test p-s pairs (p, s) for which the correct solution s is obtained by HC from (p_0, s_0). The classification task is difficult because for some problems p, multiple geometrically meaningful solutions (all points in front of cameras, small ratios between the depths, and small baseline) exist. Therefore, we also consider MLP classifiers which return m best anchors. Then, the classification is successful if the correct solution s can be tracked from one of the selected anchors.

We compare our classifier to the following baselines:

- B1 ($m = |A|$) Start from every anchor in A. (The "oracle strategy" that emulates guessing the best anchor.)
- B2 ($m = 1$) Start from the closest anchor (p_0, s_0) in terms of Euclidean distance.
- B3 ($m = 1$) Start from the closest anchor (p_0, s_0) in terms of Mahalanobis distance.

Note that the first baseline gives the upper bound on the success rate for a given anchor set A. The downside of this baseline is that HC paths from all $|A|$ anchors must be tracked. Success rate and total time for different classifiers are shown in Tab. 4. The solution s is considered correct if the squared Euclidean distance from the obtained solution to the ground-truth solution is less than 10^{-5}.

6. Homotopy continuation

We now recall the basic principles of HC methods [7, 47, 63] in the framework of our work. Suppose we have a *square system* of n polynomial equations $f(p, s) = (f_1(p, s), \ldots, f_n(p, s))$ in n unknowns $s = (s_1, \ldots, s_n)$ that vanish on our problem/solution manifold M.

[7]The procedure for generating training data is illustrated in Figure 6.

	5pt problem			4pt problem		
	ρ [%]	$\mu_t[\mu s]$	$\epsilon_t[\mu s]$	ρ[%]	$\mu_t[\mu s]$	$\epsilon_t[\mu s]$
B1, A_{50}	47.3	∞	∞	44.2	∞	∞
B1, A_{75}	74.2	∞	∞	72.0	∞	∞
B1, A_{90}	87.7	∞	∞	87.9	∞	∞
B2, A_{50}	9.9	11.8	119.5	5.2	16.1	310.5
B2, A_{75}	9.0	12.4	137.8	4.9	16.7	340.9
B2, A_{90}	8.4	12.1	144.5	5.0	16.3	324.5
B2, A	11.2	327.9	2927.7	9.8	150.1	1531.6
B3, A_{50}	14.0	12.2	87.3	5.1	15.9	312.2
B3, A_{75}	13.4	12.8	95.3	4.8	17.0	352.7
B3, A_{90}	4.2	19.5	460.2	4.8	19.9	413.5
MLP, A_{50}	29.3	15.7	53.5	21.6	19.7	91.3
MLP, A_{75}	38.8	15.0	38.7	27.8	20.3	73.0
MLP, A_{90}	39.9	14.3	35.8	29.2	19.6	66.9
MLP$_T$ A_{50}	17.0	4.6	26.9	9.1	8.9	96.8
MLP$_T$ A_{75}	29.0	7.6	**26.1**	19.0	13.5	71.1
MLP$_T$ A_{90}	36.8	10.8	29.3	26.3	16.2	**61.6**

Table 4. Classifier evaluation. Rows correspond to start problem selection strategies. The anchors are extracted from datasets *Office* and *Terrains* (Tab. 1). The strategies are evaluated on datasets *Delivery_area* and *Facade*. A_n denotes a set of anchors covering n% of the training datasets. "B1, A_n" starts from all anchors in A_n; this is the "Oracle strategy" used in Tab. 3. "B2, A_n" starts from the closest anchor to the target problem using Euclidean distance in the space of normalized image points. "B3, A_n" starts from the closest anchor using the Mahalanobis distance. **Our** "MLP $_T$ A_n" starts from the highest-scoring anchor, using the MLP described in Sec.5. Columns: ρ is the success rate (recall) of retrieving a starting point from which the target problem can be reached, μ_t is the mean solving time, is the mean effective solving time ϵ_t (as a percentage), i.e. the average time to obtain one correct solution. If multiple anchors are selected, the classification is considered successful if any of the tracks ends in a correct solution. A solution is considered correct if the squared Euclidean distance from the obtained solution to the ground-truth solution is less than 10^{-5}. We measure total time needed to perform the classification, preprocessing of the input problem and HC from all selected anchors.

Our task is to numerically continue a known problem/-solution pair $(p_0, s_0) \in M$ to a pair $(p, s) \in M$ for some problem of interest $p \in P$. This may be accomplished by introducing a parameter homotopy $H(s, t) = f(p(t), s)$ where $p(t) : [0, 1] \to P$ is some differentiable function with $p(0) = p_0$ and $p(1) = p$. The goal is to compute a differentiable path $(p(t), s(t)) : [0, 1] \to M$ such that $s(0) = s_0$ and satisfying the implicit equation $H(p(t), s(t)) = 0$. Note that the homotopy H depends on $p(t)$, and that many choices are possible. We mainly consider **Linear segment HC**: that is, we choose $p(t) = (1 - t)\, p_0 + t\, p$.

We approximate the solution curve $s(t)$ may be computed with numerical predictor/corrector methods, illus-

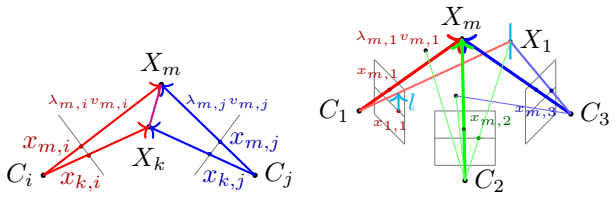


Figure 3. (left) The main geometrical constraint. (right) Scranton formulation. The world point X_1 projects under the first camera to an image point at distance l from $x_{1,1}$.

trated in Fig. 2b. In our predictor step, $s(t_i)$ for a given $t_i \in [0,1)$ is known, and $s(t_i + \Delta t)$ (for an adaptively-chosen stepsize Δt) is approximated using the standard Runge-Kutta method. In the corrector step, the value $s(t_i + \Delta t)$ is refined by up to 3 steps of Newton's method.

For both cases of 5pt and Scranton problems, there are additional polynomial constraints which rule out certain spurious solutions. However, one advantage of our HC method is that, although the vanishing set of the f we use is strictly larger than M, these *additional constraints do not need to be explicitly enforced*—See SM Sec. 16

6.1. Efficient HC implementation

Our work builds on the core of an optimized HC solver introduced in [20] that was originally developed for the problem of computing the relative pose of three calibrated cameras from corresponding point-line incidences. The optimized solver in that work is globally convergent with probability 1, but needs about 500 miliseconds to track 312 complex solution paths to solve a single problem instance.

By contrast, our solver for Scranton tracks a single path in under 10 microseconds, a speedup of more than 1000×. As noted in Sec. 1.3, much of this dramatic speedup is because global HC methods must compute all solutions over the complex numbers, whereas our method computes one, allowing now a greater probability of failure for a given data sample. Moreover, the start system in our HC method is tailored to the input by the anchor selection procedure.

There are also significant implementation-specific speedups. For instance, we obtain another $\approx 14\times$ speedup by performing all computations in *real*, instead of complex arithmetic. We also obtain an $\approx 5\times$ speedup by optimizing the linear algebra underlying predictor/corrector steps; the Jacobian matrices of our depth-formulated are sparse, leading to inexpensive closed-form solutions.[8]

7. Minimal problem formulation

We now describe the polynomial systems used by our 5pt and 4pt HC solvers. The unknowns in both systems are normalized depths in each camera, as formulated in previous works [52, 69]. The basic principle underlying both formulations, depicted in Fig. 3, is that the distances between any two 3D points must be the same, regardless of the 2D points used to reconstruct them. Throughout, we write $v_{k,i} = [x_{k,i}; 1]$ for the homogeneous coordinates of an image point. We choose these formulations over others because (i) due to low degree and sparsity, they lead to fast evaluation of straight-line programs and execution of linear algebra subroutines used in HC and (ii) they work well in tandem with our input normalization described in Sec. 8.

7.1. 5pt minimal problem

The 5pt problem is parametrized by the projection of 5 3D points into 2 calibrated cameras. There are 10 unknown depths $\lambda_{i,j}$, $i = 1,\dots,5$, $j = 1,2$, and $10 = \binom{5}{2}$ equations

$$\|\lambda_{k,1}v_{k,1} - \lambda_{m,1}v_{m,1}\|^2 = \|\lambda_{k,2}v_{k,2} - \lambda_{m,2}v_{m,2}\|^2 \quad (1)$$

with $k,m = 1,\dots,5$, $k \neq m$. To dehomogenize this system, we set $\lambda_{1,1} = 1$, obtaining a system of 10 equations in 9 unknowns. This system has *at most* 10 nonsingular solutions with all depths positive and which can be extended to a rotation with $\det R_2 = 1$. As we explain thoroughly in SM Sec. 14, there exist other, spurious solutions which satisfy both this system of equations and the *square subsystem* of 9 equations in 9 unknowns used by our HC solver. In principle, these spurious solutions may be ruled out by enforcing additional polynomial constraints. However, as already noted in Sec. 6, it is not necessary to enforce these additional constraints with our HC approch.

7.2. Scranton relaxation of the 4pt problem

The 4pt problem consists of the projection of 4 points into 3 calibrated cameras. Here we get 12 equations which are homogeneous in the 12 unknown depths:

$$\|\lambda_{k,i}v_{k,i} - \lambda_{m,i}v_{m,i}\|^2 = \|\lambda_{k,j}v_{k,j} - \lambda_{m,j}v_{m,j}\|^2 \quad (2)$$

with $k,m = 1,\dots,4$, $k \neq m$ and, e.g., $i = 1,2$, $j = i+1$. To dehomogenize the system, we set $\lambda_{1,1} = 1$. Unlike for the 5pt problem, here we get an overconstrained system of 12 equations for 11 unknown depths, which has no solution for generic (noisy) parameters $v_{k,i}$. To get a minimal problem, we replace $v_{1,1}$ by $v_{1,1} + l\,[0;1;0]$, where l is a new unknown. This relaxation allows us to "adjust" the second coordinate of the first point in the first camera. Notice that by relaxing the first point in the first view, which has $\lambda_{1,1} = 1$, the equations involving that point

$$\|v_{1,1} + l\,[0;1;0] - \lambda_{m,1}v_{m,1}\|^2 = \|\lambda_{1,2}v_{1,2} - \lambda_{m,2}v_{m,2}\|^2$$

for $m = 2,3,4$ remain quadratic in unknowns (λ, l). Solutions to the minimal problem "Scranton", Fig. 3, are solutions to this square, inhomogeneous system.[9].

[8] See SM Sec. 16 for more details.

[9] See SM Sec. 15 for additional details about Scranton.

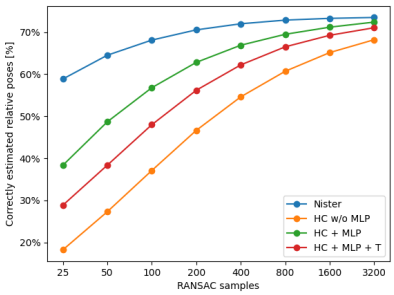


Figure 4. Percentage of camera pairs from the 2020 RANSAC Tutorial [31] for which the relative pose obtained by RANSAC has rotation and translation error less than $10°$.

8. Problem preprocessing

To simplify both learning the anchor selection strategy σ and HC tracking, we considered several schemes for *normalizing* the input image correspondences, i.e., the parameters p of the problems. Our chosen normalization yields single representative p for all problems that differ from p up to camera re-orientation or permutation of cameras or correspondences. Once the normalized problem is solved, the original problem may be solved by applying a transformation R_i^{-1} described below.

For the 5pt problem, a problem p is given by image coordinates $x_{i,j} \in \mathbb{R}^2$ with cameras indexed by $i = 1, 2$ and points indexed by $j = 1, \dots, 5$. First, we construct unit 3D vectors representing the rays of the image points as $v_{i,j} = [x_{i,j}; 1]/\|[x_{i,j}; 1]\|$. Next, we compute the mean ray for each camera $m_i = \text{mean}_j(v_{i,j})$. Then, we find the ray $v_{i^*j^*}$ that contains the largest angle with the mean ray m_i of its camera, i.e., $(i^*, j^*) = \text{argmax}_{(i,j)} \measuredangle(x_{i,j}, m_i)$. Next, we compute $w_{i,j} = R_i v_{i,j}$ such that $R_i m_i = [0; 0; 1]$ and y_{i^*,j^*}, as well as the corresponding y_{1-i^*,j^*}, have the second coordinate equal to 0, i.e., we put them on the "x axis". Finally, we swap the cameras to make the camera i^* the first one, project 3D rays $w_{i,j}$ back to the image points $x_{i,j} = w_{i,j}/w_{i,j}^{(3)}$, and reorder the image correspondences counterclockwise starting with j^*[10]

For the 4pt problem, cameras are ordered according to angles of the first point; $\measuredangle([x_{1,1}; 1], [0, 0, 1]) \geq \measuredangle([x_{2,1}; 1], [0, 0, 1]) \geq \measuredangle([x_{3,1}; 1], [0, 0, 1])$.

9. Experiments - RANSAC evaluation

To show how our method generalizes to different real scenes and to data contaminated by noise and wrong matches, we evaluate our approach for the 5pt problem on the dataset from the CVPR 2020 RANSAC Tutorial [31] consising of 2 validation scenes and 11 test scenes, each comprising 4950 camera pairs. For every camera pair, a set of matched 2D points is known. The points are contaminated with noise and mismatches. We evaluate solvers by plugging them into a RANSAC scheme [53] and computing relative poses for camera pairs in each scene. We evaluate the rotation error and translation error separately. Our evaluation metric is the percentage of relative poses whose angular distance from the ground truth is less than $10°$. We believe that this metric is justified, because the main purpose of the RANSAC procedure is to separate the correct matches from the mismatches, and a more precise relative pose can be obtained by local optimization on the inliers.

We consider our HC solver with a single anchor, our HC solver with MLP without trash, and our HC solver with MLP and trash. To estimate the success rate of our solver on this data, we compare it with the Nistér 5 point solver [48]. The success rate of the Nistér solver is close to 100%, and its errors are only due to the noise and mismatches in the data. Therefore, if the success rate of a solver on the given data is, e.g., 25%, we expect it to need 4 times the number of samples used for the Nistér solver to get the same results. We have considered RANSAC with $25, 50, 100, 200, 400, 800, 1600,$ and 3200 samples. The inlier ratio is 3px. The relation between the number of samples and the percentage of correctly estimated cameras is shown in Fig. 4. The graph shows that the lower success rate of our method can be compensated by running RANSAC for more samples. Our method with MLP requires about 4 times more samples than the Nistér solver. Therefore, the success rate on the data from [31] is around 25%, which is about 1.6 times lower than the success rate on the testing data from ETH 3D dataset [11].

10. Conclusion

Our approach to solving hard minimal problems for RANSAC framework, which uses efficient homotopy continuation and machine learning to avoid solving for many spurious solutions, is fast and delivers correct results. Supplementary Material (SM) presents more details and experiments. Our code and and data are available at https://github.com/petrhruby97/learning_minimal.

Limitations of our approach: First, we sacrifice the high success rate of a complex HC method for a fast, real HC method that fails more frequently. Nevertheless, when combined with trained models, our method succeeds in computing real solutions often enough to be useful in RANSAC. Secondly, our MLP model represents only what it is trained for. Still, we saw that it was able to represent real data distributions while keeping small size and fast evaluation. Fitting to a particular data distribution may also be useful in special situations, e.g., when cameras are mounted on a vehicle, hence having special motions.

[10] See SM Fig. 8 and Sec. 17 for more details.

[11] See SM Sec. 18 for the study of our engineering choices.

References

[1] Sameer Agarwal, Hon-leung Lee, Bernd Sturmfels, and Rekha R. Thomas. On the existence of epipolar matrices. *International Journal of Computer Vision*, 121(3):403–415, 2017. 1

[2] Chris Aholt and Luke Oeding. The ideal of the trifocal variety. *Math. Comput.*, 83(289):2553–2574, 2014. 2

[3] Hatem Said Alismail, Brett Browning, and M Bernardine Dias. Evaluating pose estimation methods for stereo visual odometry on robots. In *the 11th International Conference on Intelligent Autonomous Systems (IAS-11)*, January 2011. 1

[4] Daniel Barath. Five-point fundamental matrix estimation for uncalibrated cameras. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 235–243, 2018. 1

[5] Daniel Barath and Levente Hajder. Efficient recovery of essential matrix from two affine correspondences. *IEEE Trans. Image Processing*, 27(11):5328–5337, 2018. 1

[6] Daniel Barath, Tekla Toth, and Levente Hajder. A minimal solution for two-view focal-length estimation using two affine correspondences. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2557–2565, 2017. 1

[7] Daniel J. Bates, Andrew J. Sommese, Jonathan D. Hauenstein, and Charles W. Wampler. *Numerically Solving Polynomial Systems with Bertini*, volume 25 of *Software, environments, tools*. SIAM, 2013. 2, 6

[8] Snehal Bhayani, Zuzana Kukelova, and Janne Heikkilä. A sparse resultant based method for efficient minimal solvers. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 1767–1776. IEEE, 2020. 2

[9] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 3

[10] Paul Breiding and Sascha Timme. Homotopycontinuation.jl: A package for homotopy continuation in julia. In James H. Davenport, Manuel Kauers, George Labahn, and Josef Urban, editors, *Mathematical Software - ICMS 2018 - 6th International Conference, South Bend, IN, USA, July 24-27, 2018, Proceedings*, volume 10931 of *Lecture Notes in Computer Science*, pages 458–465. Springer, 2018. 2

[11] Martin Byröd, Klas Josephson, and Kalle Åström. A column-pivoting based strategy for monomial ordering in numerical Gröbner basis calculations. In *European Conference on Computer Vision (ECCV)*, volume 5305, pages 130–143. Springer, 2008. 1

[12] F. Camposeco, T. Sattler, and M. Pollefeys. Minimal solvers for generalized pose and scale estimation from two rays and one point. In *ECCV – European Conference on Computer Vision*, pages 202–218, 2016. 1

[13] David Cox, John Little, and Donald O'Shea. *Using Algebraic Geometry*. Springer, 1998. 1

[14] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 2018. 2

[15] T. Duff, K. Kohn, A. Leykin, and T. Pajdla. PLMP - point-line minimal problems in complete multi-view visibility. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1675–1684, 2019. 2, 12

[16] Timothy Duff, Kathlén Kohn, Anton Leykin, and Tomás Pajdla. PL_1P - 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 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVI*, volume 12371 of *Lecture Notes in Computer Science*, pages 175–192. Springer, 2020. 2, 3, 12, 14

[17] Timothy Duff, Viktor Korotynskiy, Tomas Pajdla, and Margaret H Regan. Galois/monodromy groups for decomposing minimal problems in 3d reconstruction. *arXiv preprint arXiv:2105.04460*, 2021. 14

[18] Ali Elqursh and Ahmed M. Elgammal. Line-based relative pose estimation. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3049–3056. IEEE Computer Society, 2011. 1

[19] Ioannis Z. Emiris. A general solver based on sparse resultants. *CoRR*, abs/1201.5810, 2012. 2

[20] Ricardo Fabbri, Timothy Duff, Hongyi Fan, Margaret H. Regan, David da Costa de Pinho, Elias P. Tsigaridas, Charles W. Wampler, Jonathan D. Hauenstein, Peter J. Giblin, Benjamin B. Kimia, Anton Leykin, and Tomás Pajdla. TR-PLP - Trifocal relative pose from lines at points. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 12070–12080. IEEE, 2020. 2, 7, 17, 18

[21] R. Fabbri, P. Giblin, and B. Kimia. Camera pose estimation using first-order curve differential geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 16

[22] Jean-Charles Faugère, Guillaume Moroz, Fabrice Rouillier, and Mohab Safey El Din. Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities. In J. Rafael Sendra and Laureano González-Vega, editors, *Symbolic and Algebraic Computation, International Symposium, ISSAC 2008, Linz/Hagenberg, Austria, July 20-23, 2008, Proceedings*, pages 79–86. ACM, 2008. 12

[23] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 1, 2

[24] Phillip Griffiths and Joseph Harris. *Principles of algebraic geometry*. Wiley Classics Library. John Wiley & Sons, Inc., New York, 1994. Reprint of the 1978 original. 15

[25] J. A. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendungen in Geodäsie. *In Grunerts Archiv für Mathematik und Physik*, 1841. 12

[26] Robert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. Comput. Vis.*, 13(3):331–356, 1994. 12

[27] R. Hartley and Hongdong Li. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE PAMI*, 34(12):2303–2314, 2012. 1, 3

[28] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2nd edition, 2003. 12, 14

[29] Janne Heikkilä. Using sparse elimination for solving minimal problems in computer vision. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 76–84. IEEE Computer Society, 2017. 2

[30] Robert J. Holt and Arun N. Netravali. Uniqueness of solutions to three perspective views of four points. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(3):303–307, 1995. 2

[31] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *Int. J. Comput. Vis.*, 129(2):517–547, 2021. 2, 8

[32] Joe Kileel. Minimal problems for the calibrated trifocal variety. *SIAM Journal on Applied Algebra and Geometry*, 1(1):575–598, 2017. 2, 3, 14

[33] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR – IEEE Conference on Computer Vision and Pattern Recognition*, pages 2969–2976, 2011. 12

[34] L. Kneip, R. Siegwart, and M. Pollefeys. Finding the exact rotation between two images independently of the translation. In *ECCV – European Conference on Computer Vision*, pages 696–709, 2012. 1

[35] Yubin Kuang and Kalle Åström. Pose estimation with unknown focal length using points, directions and lines. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 529–536, 2013. 1

[36] Yubin Kuang and Kalle Åström. Stratified sensor network self-calibration from TDOA measurements. In *21st European Signal Processing Conference*, 2013. 1

[37] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *European Conference on Computer Vision (ECCV)*, 2008. 1, 2

[38] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to minimal problems in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 3

[39] Viktor Larsson. PoseLib - Minimal Solvers for Camera Pose Estimation, 2020. 2

[40] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[41] Viktor Larsson, Kalle Åström, and Magnus Oskarsson. Polynomial solvers for saturated ideals. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2307–2316, 2017. 2

[42] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making minimal solvers for absolute pose estimation compact and robust. In *International Conference on Computer Vision (ICCV)*, 2017. 2, 3, 12

[43] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomás Pajdla. Beyond grobner bases: Basis selection for minimal solvers. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3945–3954, 2018. 2

[44] Anton Leykin. Numerical algebraic geometry. *Journal of Software for Algebra and Geometry*, 3(1):5–10, 2011. 2, 18

[45] Pedro Miraldo, Tiago Dias, and Srikumar Ramalingam. A minimal closed-form solution for multi-perspective pose estimation using points and lines. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, pages 490–507, 2018. 1

[46] Faraz M Mirzaei and Stergios I Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *International Conference on Computer Vision (ICCV)*, 2011. 1

[47] A. Morgan. *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009. 6

[48] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, June 2004. 1, 2, 3, 8

[49] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition (CVPR)*, pages 652–659, 2004. 1

[50] David Nistér and Frederik Schaffalitzky. Four points in two or three calibrated views: Theory and practice. *International Journal of Computer Vision*, 67(2):211–231, 2006. 2, 3

[51] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (P3P) solver. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11208 of *Lecture Notes in Computer Science*, pages 334–349. Springer, 2018. 12

[52] Long Quan, Bill Triggs, and Bernard Mourrain. Some results on minimal euclidean reconstruction from four points. *Journal of Mathematical Imaging and Vision*, 24(3):341–348, 2006. 2, 3, 7, 19

[53] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. USAC: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 35(8):2022–2038, 2013. 1, 2, 8

[54] S. Ramalingam and P. F. Sturm. Minimal solutions for generic imaging models. In *CVPR – IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1

[55] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks, 2018. 1

[56] Yohann Salaün, Renaud Marlet, and Pascal Monasse. Robust and accurate line- and/or point-based pose estimation without manhattan assumptions. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[57] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1744–1756, 2017. 1

[58] Olivier Saurer, Marc Pollefeys, and Gim Hee Lee. A minimal solution to the rolling shutter pose estimation problem. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1328–1334. IEEE, 2015. 1

[59] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3

[60] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 17

[61] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *ACM SIGGRAPH*, 2006. 1

[62] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision (IJCV)*, 80(2):189–210, 2008. 1

[63] Andrew J. Sommese and Charles W. Wampler II. *The numerical solution of systems of polynomials - arising in engineering and science*. World Scientific, 2005. 6, 14

[64] H. Stewenius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS J. of Photogrammetry and Remote Sensing*, 60:284–294, 2006. 1, 2

[65] Bernd Sturmfels. *Solving Systems of Polynomial Equations*, volume 97 of *CBMS Regional Conferences Series*. Amer.Math.Soc., Providence, Rhode Island, 2002. 1, 2

[66] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 1

[67] Jonathan Ventura, Clemens Arth, and Vincent Lepetit. An efficient minimal solution for multi-camera motion. In *International Conference on Computer Vision (ICCV)*, pages 747–755, 2015. 1

[68] Jan Verschelde. Polynomial homotopy continuation with phcpack. *ACM Commun. Comput. Algebra*, 44(3/4):217–220, 2010. 2

[69] Ji Zhang, Mireille Boutin, and Daniel G Aliaga. Pose-free structure from motion using depth from motion constraints. *IEEE transactions on image processing*, 20(10):2937–2953, 2011. 7