Learning to Solve Hard Minimal Problems
Supplementary Material

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

Here we give additional details for the main paper *P. Hruby, T. Duff, A. Leykin, T. Pajdla. Learning to Solve Hard Minimal Problems. CVPR 2022.* including an analysis of the 5pt and Scranton minimal problem solutions, details of our efficient homotopy continuation implementation, and experiments justifying our engineering choices. We also attach our code which will be made publicly available on Github.

11. A classical example of a minimal problem

A classical, easy, but still essential, minimal problem in computer vision is computing the pose of a calibrated perspective camera [28] from three points in space and their image projections [25, 26, 33, 42, 51]. In one of its classical formulations [25], it leads to a polynomial system of three equations

$$\begin{aligned}||X_1 - X_2||^2 &= ||\lambda_1 u_1 - \lambda_2 u_2||^2\\ ||X_2 - X_3||^2 &= ||\lambda_2 u_2 - \lambda_3 u_3||^2\\ ||X_3 - X_1||^2 &= ||\lambda_3 u_3 - \lambda_1 u_1||^2\end{aligned}$$

of degree two in three unknown depths $\lambda_1, \lambda_2, \lambda_3$. Parameters of the problem are three 3D points $X_i \in \mathbb{R}^3$ and homogeneous coordinates $u_i \in \mathbb{R}^2$ of three image projections, altogether on $3 \times 3 + 3 \times 2 = 15$ parameters. For generic data, the system has eight complex solutions for λ's with up to eight real solutions [22]. However, often, there are only zero, two, or four real solutions with positive λ's.

This example illustrates a typical situation occurring in minimal problem solving. The minimal problem obtained by relaxing a geometrical optimization problem, which has one optimal solution, brings in seven additional (spurious) solutions. In this case, there are always at least two real solutions corresponding to seeing the three points from the opposite sides of the plain they span.

12. Interesting hard minimal problems

Recent results [15, 16] suggest that solving minimal problems with many complex solutions is interesting. A complete classification of minimal problems for points, lines, and their incidences in the calibrated multi-view geometry appeared for the case of complete multi-view visibility [15]. It has been found that there are only 30 minimal problems in that setting, but it also became clear that problems involving more than two cameras are hard for the current symbolic-numeric and homotopy continuation solvers. The number of solutions for three views starts with 64, interesting cases have 200+ solutions, and 5-view cases have 10000+ solutions. Allowing for occlusion or missed detection in images leads to even harder problems. The follow-up work [16] developed a complete classification of minimal problems for generic arrangements of points and lines in space, observed partially by three calibrated perspective cameras when each line is incident to at most one point. It has been found that there is an infinite number of such minimal problems arranged into 74575 equivalence classes when caring only about camera configurations. Interestingly, this classification involves all calibrated trifocal geometry of computer vision for nonincident points and lines in space. Out of 74575 classes, only 759 classes have less than 300 solutions. The rest have (many) more solutions. Thus, for many interesting and potentially practical problems, computing *all* solutions in a reasonable time is a task beyond the reach of current symbolic-numeric and homotopy continuation solving methods.

13. Examples of problem-solution manifolds worked out in detail

Let us now provide a detailed explanation of the problem-solution manifold concept introduced in Sec. 2.1

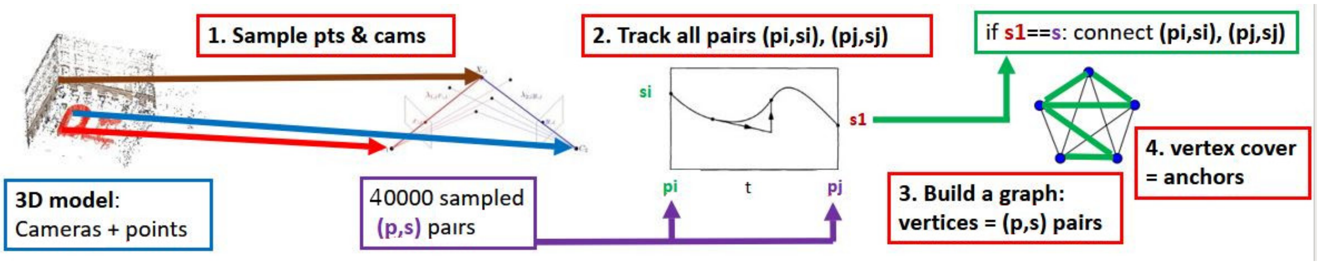
1. Sample pts & cams
2. Track all pairs (pi,si), (pj,sj)
if s1==s: connect (pi,si), (pj,sj)
si
s1
pi
t
pj
4. vertex cover = anchors
3. Build a graph: vertices = (p,s) pairs
3D model: Cameras + points
40000 sampled (p,s) pairs

Figure 5. Illustration of generating anchors. A minimal sample of cameras and points is sampled from an existing 3D model. Then, the sampled geometry is converted to the problem-solution pairs. A graph is built whose nodes are the sampled problem-solution pairs. HC is tracked from every p-s pair to every other p-s pair. If the obtained solution is equal to the sampled solution, the p-s pairs are connected with an edge. The selected anchors are a vertex cover of the graph obtained with a greedy algorithm.

for the 5pt and 4pt problems solved in this work.

Example 3. *Consider the 5pt problem of computing the relative pose of two calibrated cameras from 5 correspondences in two images,i.e., two* 3×4 *matrices* $C_1 = \begin{bmatrix} R_1 & t_1 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix}$, $C_2 = \begin{bmatrix} R_2 & t_2 \end{bmatrix}$ *in the special Euclidean group* $\mathrm{SE}_{\mathbb{R}}(3)$*, which view five world points* $X_1, \dots, X_5 \in \mathbb{R}^3$ *where* X_1 *is normalized to lie in the first image plane:* $\{X_1 \mid X_1^{(3)} = 1\} \cong \mathbb{R}^2$. *The points are in front of both cameras iff their depths*

$$\lambda_{i,j} = \lambda_{i,j}(X, C) = R_j{}^{(3,:)} X_i + t_j^{(3)}$$

are all positive ($R_j{}^{(3,:)}$ *is the third row of* R_j *and* $t_j^{(3)}$ *the third entry of* t_i*.) Consider*

$$\begin{array}{ccccccc} \Psi^{5pt} : & \left(\mathbb{R}^2 \times (\mathbb{R}^3)^4\right) & \times & \mathrm{SE}_{\mathbb{R}}(3) & \to & \left(\mathbb{R}^2\right)^{10} \times \mathbb{R}^9 \\ & (X & , & C) & \mapsto & (x, \lambda) \end{array}$$

where

$$x = \left(\lambda_{i,j}^{-1} \left(R_j X_i + t_j\right)^{(1:2)}\right)_{i=1,\dots,5;\ j=1,2;\ (i,j)\neq(1,1)}. \tag{3}$$

Here the problem space is $P = \left(\mathbb{R}^2\right)^{10}$ *the solution space is* $\mathcal{S} = \mathbb{R}^9$, $\pi(x, \lambda) = x$, *and our problem-solution manifold* $M = M^{5pt}$ *is the set of smooth points in the semialgebraic set* $\mathrm{im}(\Psi^{5pt}) \cap \left((\mathbb{R}_{>0})^9 \times \left(\mathbb{R}^2\right)^{10}\right)$.

Remark 1. *For a generic problem* $x \in M^{5pt}$*, the fiber* $\pi^{-1}(x)$ *consists of at most* 10 *solutions* $\lambda > 0$*, and every such* λ *can be extended uniquely to a pair* $(X, C) \mapsto (x, \lambda)$.

Remark 2. *Our assumptions in Example 3 imply that* $\lambda_{1,1} = 1$. *In subsequent sections, we treat the five-point problem as a system of equations in the nine remaining unknown depths. More generally, we could dehomogenize our system by setting any linear form in* λ*'s equal to* 1.

Example 4. *Consider the Scranton relaxation of the 4pt problem computing the relative pose of three calibrated cameras from 4 correspondences in 3 images. We have 4 world points* $X_1, \dots, X_4$ *with* $X_1^{(3)} = 1$ *and three cameras* $C_1 = \begin{bmatrix} I & 0 \end{bmatrix}$, $C_2 = \begin{bmatrix} R_2 & t_2 \end{bmatrix}$, $C_3 = \begin{bmatrix} R_3 & t_3 \end{bmatrix}$ *such that cameras* C_1, C_2, C_3 *view* $X_2, \dots, X_4$*, cameras* C_2, C_3 *view* X_1*, and camera* C_1 *views the line* ℓ *in the direction* $e_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top$ *that passes through* X_1*, parametrized as* $\ell(l) = X_1 + l e_2$. *Now, we have a map*

$$\begin{array}{ccccccc} \Psi^{Scr} : & \left(\mathbb{R}^2 \times (\mathbb{R}^3)^3 \times \mathbb{R}\right) & \times & \left(\mathrm{SE}_{\mathbb{R}}(3)\right)^2 & \to & \left(\mathbb{R}^2\right)^{12} \times \mathbb{R}^{12} \\ & ((X, l) & , & C) & \mapsto & (x, (\lambda, l)) \end{array}$$

where $x_{i,j}$ *are as in Eq. (3) except that*

$$x_{1,1} = \left(X_i + t_j - l e_2\right)^{(1:2)}.$$

Here the problem space is $P = \left(\mathbb{R}^2\right)^{12}$ *the solution space is* $\mathcal{S} = \mathbb{R}^{12}$, $\pi(x, \lambda) = x$, *and our problem-solution manifold* $M = M^{Scr}$ *is the set of smooth points in the semialgebraic set* $\mathrm{im}(\Psi^{Scr}) \cap \left((\mathbb{R}_{>0})^{11} \times \mathbb{R} \times \left(\mathbb{R}^2\right)^{12}\right)$.

14. Additional details for 5pt formulation

Here we provide additional details concerning the solutions of the depth-formulated 5pt problem developed in Section 7.1. Recall the system (1) of 10 equations in 9 unknown normalized depths. It has 80 solutions for generic parameters $v_{i,j}$. There are two isolated singular solutions $\lambda = [1, 0, 0, 0, 0; \pm a, 0, 0, 0, 0]$, with multiplicity 20, and 40 isolated nonsingular solutions. Among the 40 nonsingular solutions, there are at most 10 with all depths positive which extend to a rotation with $\det R_2 = 1$.

Given $(x, \lambda) \in M^{5pt}$, we first note that a valid rotation matrix $R(x, \lambda)$ may be estimated by computing certain auxiliary quantites: for $i \in \{1, \dots, 5\}$ and $v \in \{1, 2\}$, we define $X_{i,v} = \lambda_{i,v} x_{i,v}$, and, for distinct $i, j, k \in \{2, 3, 4, 5\}$,

$$A_{i,j,k}^{(v)} = \left(\begin{array}{c|c|c} X_{i,v} - X_{1,v} & X_{j,v} - X_{1,v} & X_{k,v} - X_{1,v} \end{array} \right).$$

Thus, $\det A_{i,j,k}^{(v)}$ gives the oriented volume of a tetrahedron whose vertices are $X_{1,v}, X_{i,v}, X_{j,v}, X_{k,v}$. To estimate the rotation from a geometrically meaningful solution, one may compute either

$$R_2(x, \lambda) = A^{(2)}_{2,3,4} \left(A^{(1)}_{2,3,4}\right)^{-1} \tag{4}$$

or

$$R_2(x, \lambda) = A^{(2)}_{2,3,5} \left(A^{(1)}_{2,3,5}\right)^{-1}. \tag{5}$$

Not all solutions to Equation (1) satisfy the additional constraint $\det R_2(x, \lambda) = 1$, since there is a sign-symmetry $\lambda_{i,v} \mapsto (-1)^{v+1} \lambda_{i,v}$ which changes the sign of $\det R_2(x, \lambda)$ but leaves Equation (1) invariant. Furthermore, the square subsystem obtained by dropping one of these equations has 160 solutions, counted with multiplicity. The two singular solutions $\lambda = [1, 0, 0, 0, 0; \pm a, 0, 0, 0, 0]$ now have multiplicity 32 and the number of nonsingular solutions rises to 96. Among these 96 nonsingular solutions, 76 are spurious. This set of spurious solutions includes 20 spurious solutions to the 10 equations with $\det R_2 = 1$. For each of the 76 spurious solutions, either of the matrices in Eq. (4) or Eq. (5) may have determinant -1. Thus, they may be ruled out by enforcing $\det = 1$ for both of these matrices; alternatively, we may enforce this constraint for one matrix and the dropped equation from Equation (1).

The preceding arguments illustrate that the 5pt problem, regardless of the particular equations used, has at most 20 geometrically solutions. Moreover, there is an additional symmetry given by the "twisted pair" [28, Fig. 9.12]: letting

$$t(x, \lambda) = \lambda_{1,2} v_{1,2} - R(x, \lambda) \lambda_{1,1} v_{1,1},$$

we define $tw(x, \lambda)$ coordinate-wise fixing x and

$$\mathrm{tw}(\lambda_{i,j}) = \frac{(-1)^{j+1} \|t(x, \lambda)\|^2 \lambda_{i,j}}{\|\lambda_{i,2} v_{i,2}\|^2 - \|\lambda_{i,1} v_{i,1}\|^2}.$$

The map on p-s pairs $(x, \lambda) \mapsto (x, \mathrm{tw}(x, \lambda))$ reverses the signs of depths in the second view. This justifies our claim that there are at most 10 geometrically meaningful solutions to Eq. (1). We remark that the partition of 20 solutions into twisted pairs is preserved along non-singular solution curves computed by our HC method.

15. Additional details for Scranton formulation

Unlike the system used for the 5pt problem, the square system for Scranton has infinitely many solutions. Recall that this system is given by the relaxed depth constraint

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

and Eq. (2) for remaining points and cameras. A one-dimensional family of solutions may be obtained by setting all depths except $\lambda_{1,1}, \lambda_{1,2}, \lambda_{1,3}$ to 0, resulting in 2 nontrivial equations in the remaining 3 unknowns: namely,

$$||v_{1,1} + l\,[0; 1; 0]||^2 = ||\lambda_{1,2} v_{1,2}||^2$$
$$||\lambda_{1,2} v_{1,2}||^2 = ||\lambda_{1,3} v_{1,3}||^2$$

The square system for Scranton also has several families of isolated singular solutions where certain depths equal 0. However, for generic data, the number of *nonsingular* solutions equals the number of solutions with *nonzero depths*, which is 1408. Among these, there is a four-fold sign symmetry where $\lambda_{i,2} \mapsto \pm\lambda_{i,2}, \lambda_{i,3} \mapsto \pm\lambda_{i,3}$, and $320 = 4 \times 80$ cannot be lifted to a valid pair of rotations $(R_2(x, \lambda), R_3(x, \lambda))$.

Taking these facts into account, there are at most $1408 - 3 \times 272 - 4 \times 80 = 272$ geometrically relevant solutions on the problem-solution manifold. This agrees with the number of solutions reported in both [32] and [16], where formulations in terms of trifocal tensors and camera matrices, respectively, were employed. We note that, unlike the five-point problem, there is no further reduction in the number of solutions implied by a symmetry such as the twisted pair; this follows by numerically computing the Galois group associated to Scranton, using techniques described in [17], which turns out to be the symmetric group on 272 letters.

16. Additional details on HC methods

As noted in Sec. 6, our homotopy H depends on the choice of a path $p(t)$ connecting p_0 to p, where (p_0, s_0) is a known p-s pair and p is the problem to be solved. In all of our experiments, we consider one of two choices. Mostly, we use **1) Linear segment HC**: that is, we choose $p(t) = (1 - t)\, p_0 + t\, p$. This linear segment homotopy has several advantages; among them, the straight-line programs needed to evaluate H and its derivatives are much simpler than for other paths, and the fact that $p(t)$ is real-valued for all t. However, under Linear segment HC, a differentiable solution curve $s(t)$ satisfying $H(p(t), s(t)) = 0$ need not exist for all $t \in [0, 1]$. For instance, a problem with singular solutions may exist somewhere along the segment connected in P connecting p_0 and p. However, the solution curve $s(t)$ will exist for all $t \in [0, 1]$ if p_0 and p are "close enough"—more precisely, if $p(t)$ avoids the lower-dimensional set of critical values of π in P for all $t \in [0, 1]$.

Alternatively, one may consider **2) Circular arc HC**: here, we reparametrize the segment $p(t)$ via a circular arc $t(\tau) : [0, 1] \to [0, 1]$ obtained by fixing a random $\gamma \in \mathbb{C}$ (typically of modulus 1) and $t(\tau) = \frac{\gamma\,\tau}{1 + (\gamma - 1)\tau}$. This is the γ-trick of [63, Lemma 7.1.3]. Numerical continuation with the resulting homotopy $H(s, \tau)$ is *globally convergent with probability one*: for almost all choices of γ, the solution curves $s(t)$ are defined for all $t \in [0, 1]$, and any isolated target solution (p, s) is the endpoint of some solution curve.

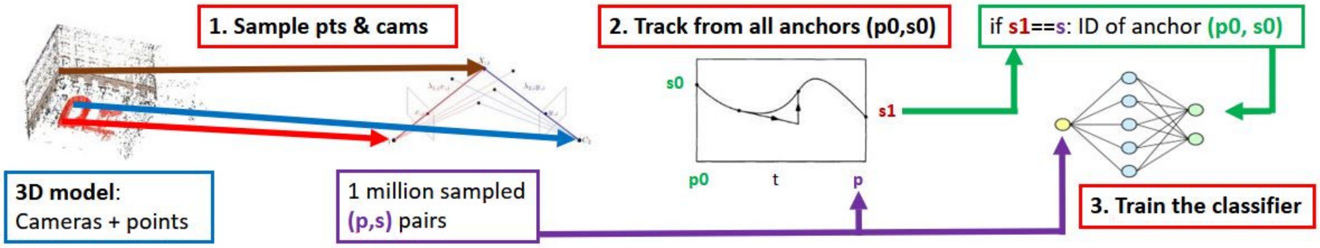
1. Sample pts & cams
2. Track from all anchors (p0,s0)
if s1==s: ID of anchor (p0, s0)
s0
s1
p0
t
p
3D model: Cameras + points
1 million sampled (p,s) pairs
3. Train the classifier

Figure 6. Illustration of generating training data and training the classifier. A minimal sample of cameras and points is sampled from an existing 3D model. Then, the sampled geometry is converted to the problem-solution pairs. Every generated problem-solution pair (p, s) is tracked from every anchor (p_0, s_0). If the correct solution is obtained, problem p is added to the training data, whereby the ID of the anchor (p_0, s_0) is used as the expected label. Then, the MLP is trained on the generated training data.

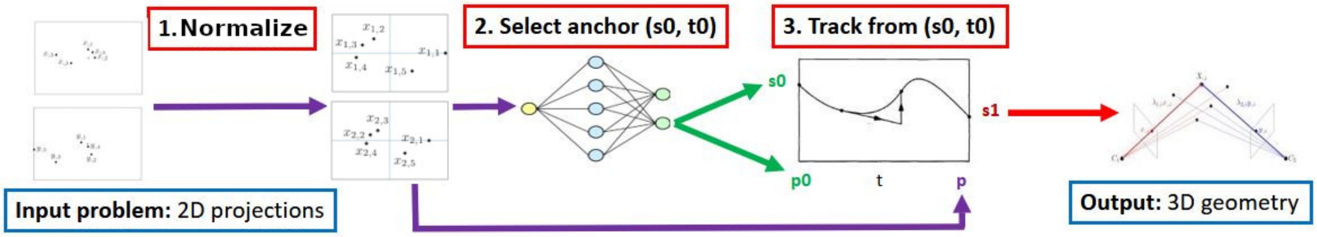

Figure 7. Illustration of testing the classifier. A minimal sample p is obtained in the RANSAC scheme. Then, the sample is normalized. The normalized sample is used as the input to the trained MLP, which selects the starting p-s pair (p_0, s_0). Then, HC is tracked from (p_0, s_0) to p. If a solution s is obtained, it is converted to a relative pose and the RANSAC score of it is evaluated.

However, this necessitates computing many spurious solutions. An experimental comparison of Linear segment and Circular arc HC may be found in Tab. 7.

We now explain why the square systems used in our homotopies are sufficient when starting from a p-s pair on the problem-solution manifold. Consider a path $t \mapsto (p(t), s(t)) \in P \times \mathcal{S}$ where $(p(0), s(0)) = (p_0, s_0) \in M$ and such that that the $n \times n$ Jacobian matrix $\frac{dH}{ds}(p(t), s(t))$ has rank n for all $t \in [0, 1]$. Thus, the path $t \mapsto (p(t), s(t))$ is contained in a single connected component of the set of nonsingular points in the complex vanishing set $\{(p, s) \in P_{\mathbb{C}} \times \mathcal{S}_{\mathbb{C}} \mid f(p, s) = 0\}$. Among these connected components is the set of smooth points in the Zariski closure of M. Indeed, the complex Zariski closure of M has a rational parametrization (given by one of the maps Ψ^{5pt}, Ψ^{Scr} defined in Sec. 13), so it is irreducible, and the connectedness of its smooth points follows by [24, pp. 21–22]. Since the point (p_0, s_0) is contained in this connected component, so also must $(p(t), s(t))$ for all $t \in [0, 1]$. Thus, any polynomial $g(p, s)$ vanishing on M satisfies $g(p(t), s(t)) = 0$ for all $t \in [0, 1]$. This means that, if HC tracking from $(p_0, s_0) \in M$ succeeds using our square system of constraints, then *all* additional constraints which are polynomial equalities are automatically satisfied. This justifies the fact that we do not explicitly enforce constraints like $\det R_2(x, \lambda) = 1$, since it is enough to enforce them for the initial p-s pair.

16.1. Efficient evaluation of predictor/corrector

The Runge-Kutta method used for the predictor step and Newton's method used for the corrector step in our HC implementation both require solving systems of linear equations. In either step, the coefficient matrix is given by the Jacobian $\frac{\partial H(s,t)}{\partial s}$ (Sec. 6). In the case of the depth formulation of the Five-Point problem (1) and the Four-Point problem (2), the associated Jacobian matrix is sparse. The sparsity pattern of the Jacobian matrix $\frac{\partial H(s,t)}{\partial s}$ is shown in (6) for the Five-Point problem, and in (7) for the Four-Point problem.

$$\begin{bmatrix} A_{0,3} & 0 & 0 & 0 & A_{0,4} & A_{0,5} & 0 & 0 & 0 \\ 0 & A_{1,1} & 0 & 0 & A_{1,4} & 0 & A_{1,6} & 0 & 0 \\ A_{2,0} & A_{2,1} & 0 & A_{2,3} & 0 & 0 & A_{2,6} & 0 & 0 \\ 0 & 0 & A_{3,2} & 0 & A_{3,4} & 0 & 0 & A_{3,7} & 0 \\ A_{4,0} & 0 & A_{4,2} & 0 & 0 & A_{4,5} & 0 & A_{4,7} & 0 \\ 0 & A_{5,1} & A_{5,2} & 0 & 0 & 0 & A_{5,6} & A_{5,7} & 0 \\ 0 & 0 & 0 & A_{6,3} & A_{6,4} & 0 & 0 & 0 & A_{6,8} \\ A_{7,0} & 0 & 0 & A_{7,3} & 0 & A_{7,5} & 0 & 0 & A_{7,8} \\ 0 & A_{8,1} & 0 & A_{8,3} & 0 & 0 & A_{8,6} & 0 & A_{8,8} \end{bmatrix} \tag{6}$$

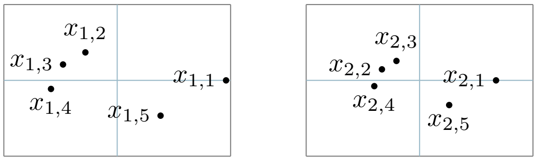

Figure 8. An example of a 5pt problem after normalization of image coordinates.

$$\begin{bmatrix} A_{0,0} & 0 & 0 & A_{0,3} & A_{0,4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{1,1} & 0 & A_{1,3} & 0 & A_{1,5} & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{2,0} & A_{2,1} & 0 & 0 & A_{2,4} & A_{2,5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{3,2} & A_{3,3} & 0 & 0 & A_{3,6} & 0 & 0 & 0 & 0 & 0 \\ A_{4,0} & 0 & A_{4,2} & 0 & A_{4,4} & 0 & A_{4,6} & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{5,1} & A_{5,2} & 0 & 0 & A_{5,5} & A_{5,6} & 0 & 0 & 0 & 0 & 0 \\ A_{6,0} & 0 & 0 & 0 & 0 & 0 & 0 & A_{6,7} & A_{6,8} & 0 & 0 & 0 \\ 0 & A_{7,1} & 0 & 0 & 0 & 0 & 0 & A_{7,7} & 0 & A_{7,9} & 0 & 0 \\ A_{8,0} & A_{8,1} & 0 & 0 & 0 & 0 & 0 & 0 & A_{8,8} & A_{8,9} & 0 & 0 \\ 0 & 0 & A_{9,2} & 0 & 0 & 0 & 0 & A_{9,7} & 0 & 0 & A_{9,10} & A_{9,11} \\ A_{10,0} & 0 & A_{10,2} & 0 & 0 & 0 & 0 & 0 & A_{10,8} & 0 & A_{10,10} & A_{10,11} \\ 0 & A_{11,1} & A_{11,2} & 0 & 0 & 0 & 0 & 0 & 0 & A_{11,9} & A_{11,10} & A_{11,11} \end{bmatrix} \tag{7}$$

Using generic methods such as LU decomposition, as in previous work [21], numerical linear algebra becomes a significant bottleneck in both the predictor and corrector stages. To overcome this bottleneck, we replace the generic numerical linear algebra with closed-form solutions to the systems of linear equations with coefficient matrices (6) and (7). This replacement results in about 5x speedup for both problems.

17. Variations of the normalization

Let us provide additional details about our normalization of problems to simplify their variability and thus to make learning of the picking function σ easier.

Fig. 8 shows an example of the normalized 5pt problem. The mean direction vectors m_1, m_2 in both images are at $[0; 0; 1]$. The first correspondence, $x_{1,1}, x_{2,1}$ is on the x axis. The first image is chosen such that it contains the larger angle with its corresponding m_u. This also means that the first correspondence point has a larger x image coordinate: $x^{(1)}_{1,1} >, x^{(1)}_{2,1}$. To make the normalized problems independent on the ordering of the correspondences, we sort them by their polar angles in the coordinate system in the the first image. Notice that their order may be swapped in the second image, e.g., as for $x_{2,2}$, $x_{3,2}$. Such a swap is mainly due to a large change of the order of depth of the corresponding points in the scene, as seen from different view points, which is in practice much less frequent than keeping the order [67].

Our normalization is chosen as the best one among several meaningful alternatives. The evaluation of the alternative normalization methods is shown in Tab. 5 for the 5pt problem and in Tab. 6 for the Scranton problem. The tables show that our strategy, labeled by A, which rotates the center of mass to zero and the farthest point on the x-axis, has the best success rate for both problems. Note that every normalization strategy performs better than when tracking without normalization.

Strategy	Succ. rate	Time inv.	Time HC
No inv.	0.53%	0	14.25 μs
A.	**3.73%**	**0.43** μs	11.80 μs
B.	3.70%	0.77 μs	11.88 μs
C.	2.29%	0.36 μs	11.19 μs
D.	1.56%	1.03 μs	12.18 μs
E.	0.71%	0.68 μs	10.44 μs

Table 5. Evaluation of the normalization for the 5 pt problem. We have generated 4000 problem-solution pairs, normalized them with a given strategy and tracked HC from every p-s pair to every other. We consider strategies from Sec. 17. We measure the success rate, average time of the normalization and of HC. The track is considered successful if the squared Euclidean distance from the obtained solution to the ground-truth is less than 10^{-5}.

The normalization strategies are as follows:

A. Rotate the center of mass to zero, rotate the point farthest from zero to x-axis.
B. Rotate the center of mass to zero with an iterative procedure, rotate the point farthest from zero to x-axis.
C. Rotate the closest point to center of mass to zero and the point farthest from zero to x-axis.
D. Rotate the center of mass to zero and the maximal variance to x-axis.
E. Rotate the closest point to center of mass to zero and the maximal variance to x-axis.

In the case of the Scranton problem, we also have to decide which point in which view to relax on the line. Here, we consider:

a) The farthest point and view.
b) The point rotated to zero (if possible).

Our normalization induces three linear constraints for every view. The instance p of the 5pt problem consists of 2D projections of 5 points into two views, therefore $p \in \mathbb{R}^{20}$. The normalized instances live in a $20 - 2 \times 3 = 14$ dimensional subspace of $\mathbb{R}^{20}$. The instance p of the 4pt problem consists of 2D projections of 4 points into 3 views, $p \in \mathbb{R}^{24}$. The normalized instances live in a $24 - 3 \times 3 = 15$ dimensional subspace of $\mathbb{R}^{24}$.

The values of the success rate in this experiment are low because we use randomly sampled data and track from every p-s pair to every other p-s pair. Tab. 9 shows that the success rate significantly increases if we track from preselected anchors which are chosen to perform well and we select the best starting anchor with a trained classifier.

Strategy	Line strategy	Succ. rate	Time inv.	Time HC
No inv.	-	0.21%	0	22.11 μs
A.	a)	**1.44%**	**0.50** μs	17.82 μs
B.	a)	**1.44%**	1.16 μs	17.40 μs
C.	a)	0.80%	0.82 μs	19.57 μs
C.	b)	0.32%	0.78 μs	17.37 μs
D.	a)	0.61%	1.39 μs	20.13 μs
E.	a)	0.37%	1.01 μs	20.43 μs
E.	b)	0.27%	0.93 μs	18.46 μs

Table 6. Evaluation of the normalization for the Scranton problem. We have generated 4000 problem-solution pairs, normalized them with a given strategy and tracked HC from every p-s pair to every other. We consider strategies from Sec. 17. We measure the success rate, average time of the normalization and of HC. The track is considered successful if the squared Euclidean distance from the obtained solution to the ground-truth is less than 10^{-5}.

18. Study to justify engineering choices

Let us describe the data sets we use to study our engineering choices.

Training data set D^{5pt} consists of 40000 p-s pairs. We randomly sample pairs of cameras and 5-tuples of 3D points from the ETH 3D dataset [60] "Office" and "Terrains". Problem parameters p are 10D vectors of 2D image coordinates obtained by projecting the sampled 5-tuples of 3D points by the camera pairs. The corresponding solutions s are 10D vectors of the depths of the 3D points in the two cameras normalized to have the first depth equal to 1. Data set D^{Scr}, consisting of 40000 p-s pairs, is constructed analagously by sampling 4-tuples of 3D points and triplets of cameras. Data sets are used to select anchor sets and to train our model for selecting the starting problem-solution pair.

Anchor sets $A^{5pt}_{50}, A^{5pt}_{75}, A^{5pt}_{90}, A^{5pt}_{100}$ are selected by the procedure described in Sec. 4 such that $A^{5pt}_{50} \subset A^{5pt}_{75} \subset A^{5pt}_{90} \subset A^{5pt}_{100} \subset D^{5pt}$ where A^{5pt}_{50} of 8 anchors covers 50% of problems in D^{5pt}, A^{5pt}_{75} of 26 anchors covers 75% of problems in D^{5pt}, A^{5pt}_{90} of 70 anchors covers 90% of problems in D^{5pt} and A^{5pt}_{100} of 465 anchors covers 100% of problems in D^{5pt}. Data sets, A^{Scr}_{50}, of 16 anchors, A^{Scr}_{75}, of 50 anchors, A^{Scr}_{90}, of 134 anchors and A^{Scr}_{100}, of 1205 anchors, are constructed analagously from A^{Scr}.

Test data set V^{5pt} consists of 60000 p-s pairs constructed as above from the ETH 3D dataset "Delivery area" and "Facade". Data set V^{Scr}, consisting of 60000 p-s pairs, is constructed analagously. We use V^{5pt} and V^{Scr} in the experiments studying anchor set selection methods reported in Tab. 4.

18.1. Comparison of different tracking approaches

Data set P^{5pt} consists of of 3751 p-s pairs sampled from the ETH 3D dataset "Courtyard". A^{5pt} and P^{5pt} are disjoint. All problems in P^{5pt} are checked to be generic and can be used as good starting problem-solution pairs. We select 20 random pairwise disjoint $P^{5pt}_i \subset P^{5pt}$ consisting of 50 problem-solution pairs. Data sets P^{Scr}, consisting of 5727 problem-solutions pairs, and P^{Scr}_i, consisting of 50 p-s pairs, are constructed analagously. We use P^{5pt}_i's and P^{Scr}_i in the experiments studying variations of the homotopy continuation methods reported in Tab. 7. In this table, we measure the success rate for a given subset P_i as a percentage of different pairs $p_{i,j} \in P_i, p_{i,k} \in P_i$ for which the fabricated solution to the target problem $p_{i,k}$ can be recovered when tracking from $p_{i,j}$ to $p_{i,k}$. Then, we find the mean success rate μ_s, and the standard deviation δ_s over all subsets $P_i, i \in \{1, ..., 20\}$.

Tab. 7 shows that for every setting, our evaluation (Sec. 16.1) brings about 5x speedup over the previous work [20] without any impact on the success rate.

The table also shows that Homotopy Continuation in the complex domain tracked from every solution has almost 100% success rate, but the running time of the solver is prohibitively slow to be used in the RANSAC scheme. We can see that reducing the number of tracks, as well as tracking in $\mathbb{R}$ instead of $\mathbb{C}$ can significantly reduce the running time (about 10000x for the Scranton problem) at the cost of a lower success rate. Note (Tab. 9), that the issue with the low success rate may be remedied by selecting an appropriate starting problem-solution pair (p_0, s_0) for every target problem p.

Our "Pick and Solve" method can use various approaches for tracking from the picked starting solution. Apart from the Homotopy Continuation, we have considered the Newton method, which, according to Tab. 7 may be a promising approach, as it has a higher success rate and lower running time than Homotopy Continuation with the same setting. Another interesting approach is to optimize the original overconstrained problem with the Gauss-Newton method. We have compared these tracking approaches: For all three tracking approaches, we have found starting p-s pairs (Sec. 4) and trained the MLP to pick the best starting p-s pair. Then, we have measured the time and success rate. The comparison is shown in Tab. 8. We can see that the effective time (the average time needed to obtain one correct solution in the RANSAC scheme) of the solver using Homotopy Continuation is about 2x lower than the effective time of the solvers using the Newton method, and the Gauss-Newton method. The solver using the Homotopy Continuation and the MLP classifier has the highest success rate. The possible explanation for this is that the Newton method behaves "more randomly" than the Homotopy Continuation, and therefore, it is more difficult to train.

	Succ. rate $\mu_s \pm \delta_s$ [%] / Time μ_t [μs]		
	5pt problem		
Solving technique	M2 [44]	MINUS [20]	OUR
$\mathbb{C}$-HC, All Sols	$98.9 \pm 0.2/ 1.9 \times 10^5$	$97.7 \pm 0.2 / 1.5 \times 10^4$	$97.7 \pm 0.1 / 5.1 \times 10^3$
$\mathbb{C}$-HC, $\mathbb{R}$ Sols	$56.1 \pm 3.3 / 1.2 \times 10^5$	$55.3 \pm 2.6 / 5.7 \times 10^3$	$54.7 \pm 3.2 / 1.9 \times 10^3$
$\mathbb{C}$-HC, Fab Sol	$12.9 \pm 0.9 / 9.8 \times 10^4$	$12.0 \pm 1.5 / 6.4 \times 10^2$	$13.1 \pm 1.4 / 1.7 \times 10^2$
$\mathbb{R}$-HC, $\mathbb{R}$ Sols	$9.9 \pm 1.7 / 5.7 \times 10^4$	$9.7 \pm 1.5 / 6.5 \times 10^2$	$9.7 \pm 1.5 / 1.1 \times 10^2$
$\mathbb{R}$-HC, Fab Sol	$3.4\pm 1.3 / 4.4 \times 10^4$	$2.7 \pm 0.8 / 6.8 \times 10^1$	$2.7 \pm 0.8 / 1.1 \times 10^1$
Newton, Fab Sol	$4.0\pm 0.6 / 1.4 \times 10^4$	$4.0 \pm 0.7 / 8.5 \times 10^0$	$4.0 \pm 0.7 / 1.3 \times 10^0$

	Succ. rate $\mu_s \pm \delta_s$ [%] / Time μ_t [μs]	
	Scranton	
Solving technique	MINUS [20]	OUR
$\mathbb{C}$-HC, All Sols	$95.5 \pm 2.6 / 6.1 \times 10^5$	$95.7 \pm 2.6 / 1.9 \times 10^5$
$\mathbb{C}$-HC, $\mathbb{R}$ Sols	$22.7 \pm 1.6 / 4.7 \times 10^4$	$22.5 \pm 1.7 / 1.5 \times 10^4$
$\mathbb{C}$-HC, Fab Sol	$3.5 \pm 0.7 / 1.3 \times 10^3$	$3.3 \pm 0.9 / 4.1 \times 10^2$
$\mathbb{R}$-HC, $\mathbb{R}$ Sols	$7.5 \pm 1.2 / 2.5 \times 10^3$	$7.6 \pm 1.2 / 4.8 \times 10^2$
$\mathbb{R}$-HC, Fab Sol	$1.2 \pm 0.3 / 7.0 \times 10^1$	$1.2 \pm 0.3 / 1.4 \times 10^1$
Newton, Fab Sol	$1.9 \pm 0.6 / 8.6 \times 10^0$	$2.0 \pm 0.4 / 1.4 \times 10^0$

Table 7. Homotopy continuation study. The rows represent variations mixing the solving technique of complex ($\mathbb{C}$-HC) and real ($\mathbb{R}$-HC) homotopy continuation, the Newton's local method (Newton) with starting from all solutions (All Sols), real solutions only ($\mathbb{R}$ Sols), and the fabricated solution only (Fab Sol) of a problem. The columns represent different implementations of homotopy continuation. M2 denotes the off-the-shelf implementation in Macaulay2 [44]. MINUS denotes the implementation based on [20]. OUR denotes our efficient implementation. To compare MINUS and OUR, we selected 20 subsets P_i, $i = 1, \dots, 20$, each containing 50 random problems from P^{5pt} for 5pt problem and from P^{Scr} for Scranton. All problems were normalized. For each P_i, we compute the success rate μ_{s_i} of $50^2 - 50$ of homotopy continuations from each start problem $p_{ij} \in P_i$ to each different target problem $p_{ik} \in P_i$. We consider a homotopy continuation successful if the fabricated solution of the target problem p_{ik} is among the solutions reached by the homotopy continuation within 10^{-5} Euclidean distance in the solution space of depths. We report the mean success rate $\mu_s = mean(\mu_{s_i})$ and the standard deviation $\delta_s = std(\mu_{s_i})$ over all P_i's for each implementation and mean computation times μ_t.

Method	ρ[%]	$\mu_t[\mu s]$	$\epsilon_t[\mu s]$
N3 + B1	0.01	1.73	15159.7
N3 + MLP	0.12	8.2	6811.2
N15 + B1	2.1	2.6	119.7
N15 + MLP	10.6	10.6	100.0
HC + B1	4.2	18.7	442.1
HC + MLP	26.3	16.2	**61.6**
GN + B1	11.64	20.04	172.2
GN + MLP	20.29	26.1	128.5

Table 8. Study of methods for starting problem selection and tracking for Scranton problem. The strategies are evaluated on datasets *Deilvery_area* and *Facade*. Tracking methods: 'N3': Newton method with 3 steps, 'N15': Newton method with 15 steps (this number of steps maximizes the efficient time), 'HC': Homotopy Continuation as described in Sec 6, 'GN': Gauss-Newton method optimizing the overconstrained 4pt problem. Problem selection methods: 'B1': Tracks always from the same anchor. 'MLP': selecting the starting problem as the one with the highest score given by the MLP.

Tab. 9 provides a more detailed analysis of the frequency of different results of real Homotopy Continuation tracked from the fabricated solution. We consider two different settings. In the "All pairs" setting, we track from each $p_i \in P$ to each other $p_j \in P$. Then, in the MLP setting, we select a starting p-s pair (p_0, s_0) from A_{90}, and track from p_0 to p. We measure how often we reach the fabricated solution, non-fabricated meaningful solution, a non-valid solution, and how often HC fails and does not deliver any solution. The table shows that the MLP increases the probability of reaching the fabricated solution about 10x for 5pt and 20x for Scranton. The probability of reaching another meaningful solution increases about 3x for both problems, while the probability of reaching a non-valid solution and the probability of failing decreases.

18.2. Comparison of different settings our solver

Here, we show how different settings of the solver influence the resulting success rate ρ, mean running time μ_t, and efficient time ϵ_t. We perform this study on our solver for the Scranton problem. For every experiment, we use the

	All pairs		MLP	
Result [%]	5 pt	4 pt	5pt	4pt
Fabricated sol.	3.64	1.49	38.8	29.2
1 rel. pose correct	-	0.74	-	4.53
Other meaningful	9.50	13.43	31.6	34.2
Sol. with det -1	0.00	0.00	0.00	0.00
Sol. with zeros	0.03	1.53	0.01	0.52
Negative sol.	0.52	10.86	0.83	8.54
Failed track	86.31	71.94	28.8	23.0

Table 9. Percentage of different results of real homotopy continuation. In “All pairs”, we track from each $p_i \in P$ to each other $p_j \in P$. In “MLP”, we solve each $p \in V$ by selecting a starting p-s pair (p_0, s_0) from A_{90}, and by tracking HC from p_0 to p. “Fabricated sol.” means that the fabricated solution was reached, “1 rel. pose correct” means that if we convert the obtained solution of the Scranton problem to the relative poses, then at least one of three relative poses is correct. “Other meaningful” means that a non-fabricated solution with positive depths and valid rotation matrix was reached, “Sol. with det -1” means that the matrix R is not a valid rotation. “Sol. with zeros” means that some depths are equal to 0, “Negative sol.” means that some depths are negative, and “Failed track” means that the HC track has failed and, thus, the solution has not been found.

settings from the main paper, except that one parameter is varied. The solver uses anchors A_{90}^{Scr} and it is evaluated on data V^{Scr}. Our goal is to show that we have selected the optimal settings for our solver.

We compare different methods of dehomogenizing Scranton problem in Tab. 10. This justifies our choice of fixing the first depth $\lambda_{1,1} = 1$, which gives a superior success rate compared to “symmetric” and “asymmetric” dehomogenization proposed in [52].

See Fig. 9 for a code snippet showing the structure of our MLP model. In Tab. 11, we show how the success rate ρ and running time μ_t depends on the size of the MLP that is used for the classification of the anchors. Larger networks have a higher success rate. However, the efficient time of the solver with the smaller MLP is better because the time needed for the evaluation of the MLP grows faster than the success rate.

Finally, in Tab. 12, we show how the number of HC tracks per problem influences the success rate ρ, running time μ_t, and efficient time ϵ_t. In this experiment, we use the MLP trained in 5, and we perform n tracks from n anchors with the highest score. We consider the solution successful if at least one track reaches the fabricated solution of the target problem. Much like using larger MLP as Tab. 11, such a strategy involving multiple anchors suggests a future approach to improving our solvers’ success rates. However, we note that the efficient time ϵ_t in Tab. 12 grows with the number of tracks, since the evaluation time grows faster than the success rate.

Formulation	Succ. rate
First depth fixed	**2.30 %**
Quan symmetrical [52]	1.47 %
Quan asymmetrical [52]	1.13 %

Table 10. Scranton dehomogenization study. Rows correspond to different formulations of the problems. For each method, we compute the success rate of $4000^2 - 4000$ HC calls from each starting p-s pair to each other target p-s pair. We consider the result successful if the fabricated solution of the target problem is and the result computed by HC are sufficiently close ($\leq 10^{-5}$ Euclidean distance in the solution space of depths.)

Layer size	$\rho[\%]$	$\mu_t[\mu s]$	$\epsilon_t[\mu s]$
100	27.8	20.3	**73.1**
200	31.3	30.8	98.3
500	34.7	79.0	227.6

Table 11. Study of different MLP sizes. The strategies are evaluated on datasets *Deilvery_area* and *Facade*. Scranton problem, MLP+HC, A_{75}. Rows correspond to different sizes of hidden MLP layers.

# Tracks	$\rho[\%]$	$\mu_t[\mu s]$	$\epsilon_t[\mu s]$
1	29.2	19.6	**67.0**
2	37.2	33.3	89.6
3	42.2	45.0	106.8
4	45.9	60.8	132.6
8	56.4	118.1	209.5
16	67.9	245.3	361.5

Table 12. Number of tracks study. The strategies are evaluated on datasets *Deilvery_area* and *Facade*. Scranton problem, MLP+HC, A_{90}. Rows correspond to different numbers of tracks conducted after the MLP is evaluated.

References

[67] A.L. Yuille and T. Poggio. A generalized ordering constraint for stereo correspondence. A.I. Memo 777, AI Lab, MIT, 1994.

```python
class Net(nn.Module):
    def __init__(self, anchors):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(20,100)
        self.relu1 = nn.PReLU(100, 0.25)
        self.fc2 = nn.Linear(100,100)
        self.relu2 = nn.PReLU(100, 0.25)
        self.fc4 = nn.Linear(100,100)
        self.relu4 = nn.PReLU(100, 0.25)
        self.fc5 = nn.Linear(100,100)
        self.relu5 = nn.PReLU(100, 0.25)
        self.fc6 = nn.Linear(100,100)
        self.relu6 = nn.PReLU(100, 0.25)
        self.fc7 = nn.Linear(100,100)
        self.relu7 = nn.PReLU(100, 0.25)
        self.drop3 = nn.Dropout(0.5)
        self.fc3 = nn.Linear(100,anchors+1)
    def forward(self, x):
        x = self.relu1(self.fc1(x))
        x = self.relu2(self.fc2(x))
        x = self.relu4(self.fc4(x))
        x = self.relu5(self.fc5(x))
        x = self.relu6(self.fc6(x))
        x = self.relu7(self.fc7(x))
        x = self.drop3(x)
        return self.fc3(x)
```

Figure 9. Code snippet describing our MLP.