Optimizing Elimination Templates by Greedy Parameter Search*

Evgeniy Martyushev
South Ural State University
martiushevev@susu.ru

Jana Vrablikova
Department of Algebra
MFF, Charles University
j.vrablikov@gmail.com

Tomas Pajdla
CIIRC - CTU in Prague
pajdla@cvut.cz

Abstract

We propose a new method for constructing elimination templates for efficient polynomial system solving of minimal problems in structure from motion, image matching, and camera tracking. We first construct a particular affine parameterization of the elimination templates for systems with a finite number of distinct solutions. Then, we use a heuristic greedy optimization strategy over the space of parameters to get a template with a small size. We test our method on 34 minimal problems in computer vision. For all of them, we found the templates either of the same or smaller size compared to the state-of-the-art. For some difficult examples, our templates are, e.g., 2.1, 2.5, 3.8, 6.6 times smaller. For the problem of refractive absolute pose estimation with unknown focal length, we have found a template that is 20 times smaller. Our experiments on synthetic data also show that the new solvers are fast and numerically accurate. We also present a fast and numerically accurate solver for the problem of relative pose estimation with unknown common focal length and radial distortion.

1. Introduction

Many tasks in 3D reconstruction [49, 50] and camera tracking [46, 55] lead to solving minimal problems [1, 4, 10, 25, 29, 36, 38, 45, 48, 51], which can be formulated as systems of polynomial equations.

The state-of-the-art approach to efficient solving polynomial systems for minimal problems is to use symbolic-numeric solvers based on elimination templates [5, 29, 33]. These solvers have two main parts. In the first offline part, an elimination template is constructed. The template consists of a map (formulas) from input data to a (Macaulay) coefficient matrix. The template is the same for different input generic (noisy) data. In the second online phase, the coefficient matrix is filled by the data of a particular problem, reduced by the Gauss–Jordan (G–J) elimination and used to construct an eigenvalue/eigenvector computation problem of an action matrix that delivers the solutions of the system.

While the offline phase is not time critical, the online phase has to be computed very fast (mostly in submillisecond time) to be useful for robust optimization based on RANSAC schemes [16]. Therefore, it is important to build templates (*i.e*., Macaulay matrices) that are as small as possible to make the G–J elimination fast. Besides the size, we also need to pay attention to building templates that lead to numerically stable computation.

1.1. Contribution

We develop a new approach to constructing elimination templates for efficiently solving minimal problems. First, using the general syzygy-based parameterization of elimination templates from [33], we construct a partial (but still generic enough) parameterization of templates. Then, we apply a greedy heuristic optimization over the space of parameters to find as small a template as possible.

We demonstrate our method on 34 minimal problems in geometric computer vision. For all of them, we found the templates either of the same or smaller size compared to the state-of-the-art. For some difficult examples, our templates are, *e.g*., 2.1, 2.5, 3.8, 6.6 times smaller. For the problem of refractive absolute pose estimation with unknown focal length, we have found a template that is 20 times smaller. Our experiments on synthetic data also show that the new solvers are fast and numerically accurate.

We propose a practical solver for the problem of relative pose estimation with unknown common focal length and radial distortion. All previously presented solvers for this problem are either extremely slow or numerically unstable.

1.2. Related work

Elimination templates are matrices that encode the transformation from polynomials of the initial system to polynomials needed to construct the action matrix. Knowing an ac-

*The research was supported by projects EU RDF IMPACT No. CZ.02.1.01/0.0/0.0/15 003/0000468 and EU H2020 No. 871245 SPRING. T. Pajdla is with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague.

tion matrix, the solutions of the system are computed from its eigenvectors. *Automatic generator* (AG) is an algorithm that inputs a polynomial system and outputs an elimination template for the action matrix computation.

Automatic generators: The first automatic generator was built in [29], where the template was constructed iteratively by expanding the initial polynomials with their multiples of increasing degree. This AG has been widely used by the computer vision community to construct polynomial solvers for a variety of minimal problems, *e.g.*, [6, 7, 31, 37, 43, 48, 59], see also [33, Tab. 1]. Paper [33] introduced a non-iterative AG based on tracing the Gröbner basis construction and subsequent syzygy-based reduction. This AG allowed fast constructing templates even for hard problems. An alternative AG based on using sparse resultants was recently proposed in [5]. This method, along with [36], are currently the state-of-the-art automatic template generators.

Improving stability: The standard way of constructing the action matrix from a template requires performing its LU decomposition. For large templates, this operation often leads to significant round-off and truncation errors and hence to numerical instabilities. The series of papers [9–11] addressed this problem and proposed several methods of improving stability, *e.g.*, by performing a QR decomposition with column pivoting on the step of constructing the action matrix from a template.

Optimizing formulations: Choosing a proper formulation of a minimal problem can drastically simplify finding its solutions. Paper [30] proposed the variable elimination strategy that reduces the number of unknowns in the initial polynomial system. For some problems, this strategy led to notably smaller templates [23, 35].

Optimizing templates: Much effort has been spent on speeding up the action matrix method by optimizing the template construction step. Paper [44] introduced a method of optimizing templates by removing some unnecessary rows and columns. The method [28] utilized the sparsity of elimination templates by converting a large sparse template into the so-called singly-bordered block-diagonal form. This allowed splitting the initial problem into several smaller subproblems, which are easier to solve. In paper [36], the authors proposed two methods that significantly reduced the sizes of elimination templates. The first method used the so-called Gröbner fan of a polynomial ideal for constructing templates w.r.t. all possible standard bases of the quotient space. The second method went beyond Gröbner bases and introduced a random sampling strategy for constructing non-standard bases.

Optimizing root solving: Complex roots are spurious for most problems arising in applications. Paper [8] introduced two methods of avoiding the computation of complex roots, which resulted in a significant speed-up of polynomial solvers.

Discovering symmetries: Polynomial systems for certain minimal problems may have hidden symmetries. Uncovering these symmetries is another way of optimizing templates. This approach was demonstrated for the simplest partial p-fold symmetries in [27, 32]. A more general case was recently investigated in [15]. Paper [34] proposed a method of handling special polynomial systems with a (possibly) infinite subset of spurious solutions.

The most related work: Our work is essentially based on the results of papers [5, 11, 33, 36].

2. Solving polynomial systems by templates

Here we review solving polynomial systems with a finite number of solutions by eigendecomposition of action matrices. We also show how are the action matrices constructed using elimination templates in computer vision. We build on nomenclature from [9, 12, 13].

2.1. Gröbner bases and action matrices

Here we introduce action matrices and explain how they are related to Gröbner bases.

We use $\mathbb{K}$ for a field, $X = \{x_1, \ldots, x_k\}$ for a set of k variables, $[X]$ for the set of *monomials* in X and $\mathbb{K}[X]$ for the polynomial ring over $\mathbb{K}$. Let $F = \{f_1, \ldots, f_s\} \subset \mathbb{K}[X]$ and $J = \langle F \rangle$ for the ideal generated by F. A set $G \subset \mathbb{K}[X]$ is a *Gröbner basis* of ideal J if $J = \langle G \rangle$ and for every $f \in J \setminus \{0\}$ there is $g \in G$ such that the leading monomial of g divides the leading monomial of f. The Gröbner basis G is called *reduced* if $c(g, \mathtt{LM}(g)) = 1$[1] for all $g \in G$ and $\mathtt{LM}(g)$ does not divide any monomial of $g' \in G$ when $g' \neq g$.

For a fixed monomial ordering (see SM Sec. 7), the reduced Gröbner basis is defined uniquely for each ideal. Moreover, for any polynomial ideal J, there are finitely many distinct reduced Gröbner bases, which all can be found using the *Gröbner fan* of J [36, 42].

For an ideal $J \subset \mathbb{K}[X]$, the *quotient ring* $\mathbb{K}[X]/J$ consists of all equivalence classes $[f]$ under the equivalence relation $f \sim g$ iff $f - g \in J$. If $J = \langle F \rangle$ is zero-dimensional, *i.e.*, the set of roots of $F = 0$ is finite, then $\mathbb{K}[X]/J$ is a finite-dimensional vector space. Moreover, $\dim \mathbb{K}[X]/J$ equals the number of solutions to $F = 0$, when counting the multiplicities [13].

Given a Gröbner basis G of ideal J, we can construct the *standard (linear) basis* $\mathcal{B}$ of the quotient ring $\mathbb{K}[X]/J$ as the set of all monomials not divisible by any leading monomial from G, *i.e.*, $\mathcal{B} = \{b \,:\, \mathtt{LM}(g) \nmid b, \forall g \in G\}$.

Fix a polynomial $a \in \mathbb{K}[X]$ and define the linear operator

$$T_a \colon \mathbb{K}[X]/J \to \mathbb{K}[X]/J \colon [f] \mapsto [a \cdot f].$$

Selecting a basis in $\mathbb{K}[X]/J$, *e.g.*, the standard one, allows to represent the operator T_a as a $d \times d$ matrix, where $d =$

[1] We denote the coefficient of g at m by $c(g, m)$.

$\dim \mathbb{K}[X]/J$. This matrix, which is also denoted by T_a, is called the *action matrix* and the polynomial a is called the *action polynomial*.

The action matrix can be found using a Gröbner basis G of ideal J as follows. Let $\{b_1, \ldots, b_d\}$ be a basis in the quotient ring $\mathbb{K}[X]/J$. For a given a, we use G to construct the *normal forms* of $a\, b_i$:

$$\overline{(a\, b_i)}^G = \sum_j t_{ij} b_j, \quad i = 1, \ldots, d,$$

where $t_{ij} \in \mathbb{K}$. Then, we have $T_a = (t_{ij})$.

2.2. Solving polynomial systems by action matrices

Action matrices are useful for computing the solutions of polynomial systems with a finite number d of solutions. The situation is particularly simple when (i) all solutions $p_j \in \mathbb{K}^k$, $j = 1, \ldots, d$, are of multiplicity one and (ii) the action polynomial a evaluates to pairwise different values on the solutions, *i.e.*, $a(p) \neq a(q)$ for all solutions $p \neq q$. Then, the action matrix T_a has d one-dimensional eigenspaces, and d vectors $\begin{bmatrix} b_1(p_j) & \ldots & b_d(p_j) \end{bmatrix}^\top$ of polynomials b_i evaluated at the solutions p_j, $i, j = 1, \ldots, d$, are basic vectors of the d eigenspaces [12, p. 59 Prop. 4.7]. Having one-dimensional eigenspaces leads to a straightforward method for extracting all solutions p_j. Thus, the classical approach to finding solutions to a polynomial system F with a finite number of solutions is as follows.

1. Choose an action polynomial a: Assuming that the solutions p_j are of multiplicity one, *i.e.*, the ideal $J = \langle F \rangle$ is radical [13, p. 253 Prop. 7], our goal is to choose a such that it has pairwise different values $a(p_j)$. This is always possible by choosing $a = x_\ell$, *i.e.*, a variable, after a linear change of coordinates [12, p. 59]. As we will see, such a choice leads to a simple solving method.

In computer vision, we are particularly interested in solving polynomial systems that consist of the union of two sets of equations $F = F_1 \cup F_2$ where F_1 do not depend on the image measurements (*e.g.*, Demazure constraints on the Essential matrix [14]) and F_2 depend on the image measurements affected by random noise (*e.g.*, linear epipolar constraints on the Essential matrix [39]). Then, the linear change of coordinates can be done only once in the offline phase to transform F_1. In the next, we will assume that there is a with pairwise distinct values on the solutions p_j.

2. Choose a basis $\mathcal{B}$ of $\mathbb{K}[X]/J$: There are infinitely many bases of $\mathbb{K}[X]/J$. Our goal is to choose a basis that leads to a simple and numerically stable solving method. Elements of $\mathcal{B}$ are equivalence classes represented by polynomials that are $\mathbb{K}$-linear combinations of monomials. Hence, the simplest bases consist of equivalence classes represented by monomials. It is important that $\mathbb{K}[X]/J$ has a standard monomial basis [12, 53] for each reduced Gröbner basis. In generic situations, the elements of $\mathcal{B}$ represented by monomials are equivalent to (infinitely) many different linear combinations of the standard monomials and thus provide (infinitely) many different vectors to construct (infinitely) many different bases of $\mathcal{B}$. In the following, we assume *monomial bases*, *i.e.*, the bases consisting of the classes represented by monomials.

3. Construct the action matrix T_a w.r.t. $\mathcal{B}$: Once a and $\mathcal{B}$ have been chosen, it is straightforward to construct T_a by the process described in Sec. 2.1. However, in computer vision, polynomial systems often have the same support for different values of their coefficients. Then, it is efficient [11, 57] to construct T_a by (i) building a Macaulay matrix M using a fixed procedure – *a template* – designed in the offline phase, and then (ii) produce T_a in the online phase by the G–J elimination of M [10, 29]. Our main contribution, Sec. 3 and Sec. 4, in this work is an efficient approach to constructing Macaulay matrices.

4. Computing the eigenvectors v_j, $j = 1, \ldots, d$ of T_a: Computing the eigenvectors of T_a is a straightforward task when there are d one-dimensional eigenspaces.

5. Recovering the solutions from eigenvectors: To find the solutions, it is enough to evaluate all unknowns x_l, $l = 1, \ldots, k$, on the solutions p_j. It can be done by writing unknowns x_l in the standard basis b_i as $x_l = \sum_i c_{li} b_i$. Then, $x_l(p_j) = \sum_i c_{li} b_i(p_j) = \sum_i c_{li} (v_j)_i$, where $(v_j)_i$ is the ith element of vector v_j.

2.3. Macaulay matrices and elimination templates

Let us now introduce Macaulay matrices and elimination templates.

To simplify the construction, we restrict ourselves to the following assumptions: (i) the elements of basis $\mathcal{B}$ are represented by monomials and (ii) the action polynomial a is a monomial and $a \neq 1$.

Given an s-tuple of polynomials $F = (f_1, \ldots, f_s)$, let $[X]_F$ be the set of all monomials from F. Let the cardinality $\#[X]_F$ of $[X]_F$ be n. Then, the *Macaulay matrix* $M(F) \in \mathbb{K}^{s \times n}$ has coefficient $c(f_i, m_j)$, with $m_j \in [X]_F$, in the (i, j) element: $M(F)_{ij} = c(f_i, m_j)$.

A *shift* of a polynomial f is a multiple of f by a monomial $m \in [X]$. Let $A = (A_1, \ldots, A_s)$ be an s-tuple of sets of monomials $A_j \subset [X]$. We define the *set of shifts* of F as

$$A \cdot F = \{m \cdot f_j \, : \, m \in A_j, f_j \in F\}. \tag{1}$$

Let $\mathcal{B}$ be a monomial basis of the quotient ring $\mathbb{K}[X]/\langle F \rangle$ and a be an action monomial. The sets $\mathcal{B}$, $\mathcal{R} = \{a\, b \, : \, b \in \mathcal{B}\} \setminus \mathcal{B}$, and $\mathcal{E} = [X]_{A \cdot F} \setminus (\mathcal{R} \cup \mathcal{B})$ are the sets of *basic*, *reducible* and *excessive monomials*, respectively [11].

Definition 1. Let $\overline{\mathcal{B}} = \mathcal{B} \cap [X]_{A \cdot F}$. A Macaulay matrix $M(A \cdot F)$ with columns arranged in ordered blocks $M(A \cdot F) = \begin{bmatrix} M_\mathcal{E} & M_\mathcal{R} & M_{\overline{\mathcal{B}}} \end{bmatrix}$ is called the *elimination template* for F w.r.t. a if the following conditions hold true:

1. $\mathcal{R} \subset [X]_{A \cdot F}$;
2. the reduced row echelon form of $M(A \cdot F)$ is

$$\widetilde{M}(A \cdot F) = \begin{bmatrix} * & 0 & * \\ 0 & I & \widetilde{M}_{\overline{\mathcal{B}}} \\ 0 & 0 & 0 \end{bmatrix}, \quad (2)$$

where $*$ means a submatrix with arbitrary entries, 0 is the zero matrix of a suitable size, I is the identity matrix of order $\#\mathcal{R}$ and $\widetilde{M}_{\overline{\mathcal{B}}}$ is a matrix of size $\#\mathcal{R} \times \#\overline{\mathcal{B}}$.

Theorem 1. *The elimination template is well defined,* i.e., *for any s-tuple of polynomials $F = (f_1, \dots, f_s)$ such that ideal $\langle F \rangle$ is zero-dimensional, there exists a set of shifts $A \cdot F$ satisfying the conditions from Definition 1.*

Proof. See SM Sec. 8. □

In SM Sec. 9, we provide several examples of solving polynomial systems by elimination templates.

2.4. Action matrices from elimination templates

We will now explain how to construct action matrices from elimination templates.

Given a finite set $A \subset \mathbb{K}[X]$, let $v(A)$ denote the vector consisting of the elements of A. If A is a set of monomials, then the elements of $v(A)$ are ordered by the chosen monomial ordering on $[X]$. For a set of polynomials, the order of elements in $v(A)$ is irrelevant.

Let $a \in [X]$ be an action monomial and let $M(A \cdot F) = \begin{bmatrix} M_{\mathcal{E}} & M_{\mathcal{R}} & M_{\overline{\mathcal{B}}} \end{bmatrix}$ be an elimination template for F w.r.t. a. Denote for short $M = M(A \cdot F)$ and $\mathcal{X} = [X]_{A \cdot F}$ the set of monomials corresponding to columns of M. Since M is a Macaulay matrix, $M\, v(\mathcal{X}) = 0$ represents the expanded system of equations.

It may happen that $\overline{\mathcal{B}} = \mathcal{B} \cap \mathcal{X}$ is a proper subset of $\mathcal{B}$, see Examples 1 and 4 in SM. Let us construct matrix $M'_{\mathcal{B}}$ by adding to $M_{\overline{\mathcal{B}}}$ the zero columns corresponding to each $b \in \mathcal{B} \setminus \overline{\mathcal{B}}$. Then, the template M is transformed into $M' = \begin{bmatrix} M_{\mathcal{E}} & M_{\mathcal{R}} & M'_{\mathcal{B}} \end{bmatrix}$, which is clearly a template too. Therefore, the reduced row echelon form of M' must be of the form (2). Thus we are getting

$$v(\mathcal{R}) = -\widetilde{M}'_{\mathcal{B}}\, v(\mathcal{B}). \quad (3)$$

To provide an explicit formula for the action matrix, let the set of basic monomials $\mathcal{B}$ be partitioned as $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$, where $\mathcal{B}_2 = \{a\, b \,:\, b \in \mathcal{B}\} \cap \mathcal{B}$ and $\mathcal{B}_1 = \mathcal{B} \setminus \mathcal{B}_2$. Then $v(\mathcal{B}) = \begin{bmatrix} v(\mathcal{B}_1) \\ v(\mathcal{B}_2) \end{bmatrix}$ and the action matrix can be read off as follows:

$$T_a = \begin{bmatrix} -\widetilde{M}'_{\mathcal{B}} \\ P \end{bmatrix}, \quad (4)$$

where P is a binary matrix, *i.e.*, a matrix consisting of 0 and 1, such that $v(\mathcal{B}_2) = P\, v(\mathcal{B})$.

3. Constructing parameterized templates

Let $\mathcal{B}$ be a monomial basis of $\mathbb{K}[X]/J$. We distinguish a standard basis, which comes from a given Gröbner basis of J, and a non-standard basis, which may be represented by arbitrary monomials from $[X]$. Given a polynomial $f \in \mathbb{K}[X]$, let $[f] = \sum_i c_i [b_i]$, where $b_i \in \mathcal{B}$ and $c_i \in \mathbb{K}$, be the unique representation of $[f]$ in the basis $\mathcal{B}$. Then, the polynomial $\sum_i c_i b_i$ is called the *normal form* for f w.r.t. $\mathcal{B}$ and is denoted by $\overline{f}^{\mathcal{B}}$. Let us fix the action monomial a, and construct $\overline{a\, v(\mathcal{B})}^{\mathcal{B}}$, *i.e.*, the vector of normal forms for each $a\, b_i$. If $\mathcal{B}$ is the standard basis corresponding to a Gröbner basis G, the normal form w.r.t. $\mathcal{B}$ is found in a straightforward way as the unique remainder after dividing by polynomials from G, *i.e.*, $\overline{a\, v(\mathcal{B})}^{\mathcal{B}} = \overline{a\, v(\mathcal{B})}^{G} = T_a\, v(\mathcal{B})$, where $T_a \in \mathbb{K}^{d \times d}$ is the action matrix.

Now, consider an arbitrary (possibly non-standard) basis $\mathcal{B}$. To construct the normal form for $a\, v(\mathcal{B})$ w.r.t. $\mathcal{B}$, we select a Gröbner basis G of ideal J and find the related (standard) basis $\widehat{\mathcal{B}}$. Then, we get $\overline{v(\mathcal{B})}^{G} = S\, v(\widehat{\mathcal{B}})$. As $\mathcal{B}$ is a basis, the square matrix S is invertible. We can also compute $\overline{a\, v(\widehat{\mathcal{B}})}^{G} = \widehat{T}_a v(\widehat{\mathcal{B}})$, where $\widehat{T}_a \in \mathbb{K}^{d \times d}$ is the matrix of the action operator in the standard basis $\widehat{\mathcal{B}}$. Then, we have $\overline{a\, v(\mathcal{B})}^{\mathcal{B}} = T_a v(\mathcal{B})$, where $T_a = S\, \widehat{T}_a S^{-1}$ is the matrix of the action operator in the basis $\mathcal{B}$.

Let us define

$$V = a\, v(\mathcal{B}) - T_a v(\mathcal{B}) \quad (5)$$

and compute

$$\begin{aligned} \overline{V}^{G} &= S\big[\overline{a \cdot (S^{-1} \overline{v(\mathcal{B})}^{G})}^{G} - \widehat{T}_a S^{-1}\, \overline{v(\mathcal{B})}^{G}\big] \\ &= S\big[\overline{a\, v(\widehat{\mathcal{B}})}^{G} - \widehat{T}_a v(\widehat{\mathcal{B}})\big] = 0. \end{aligned}$$

It follows that the elements of vector V belong to J. Therefore, there is a matrix $H \in \mathbb{K}[X]^{d \times s}$ such that

$$V = H v(F). \quad (6)$$

Knowing matrix H is enough for constructing an elimination template for F according to Definition 1. Equation (6) can be rewritten in the form $V = \sum_k h_k f_k$, where h_k is the kth column of H. Let $[X]_k$ be the support of h_k, $A = ([X]_1, \dots, [X]_s)$ and $A \cdot F$ be the related set of shifts. Then, the Macaulay matrix $M(A \cdot F)$ is the elimination template for F, see SM Sec. 8.

Now, we discuss how to construct the matrix H so that (6) holds true. As noted in [33], such matrix is not defined uniquely, reflecting the ambiguity in constructing elimination templates. One such matrix, say H_0, can be found as a byproduct of the Gröbner basis computation.[2]

[2]In practice, matrix H_0 can be derived by using an additional option in the Gröbner basis computation command, *e.g.*, `ChangeMatrix=>true` in Macaulay2 [18] or `output=extended` in Maple.

On the other hand, there is a simple algorithm for computing generators of the first syzygy module of any finite set of polynomials [12]. For the s-tuple of polynomials F, the algorithm outputs a matrix $H_1 \in \mathbb{K}[X]^{l\times s}$ such that $H_1 v(F) = 0$. Let

$$H = H_0 + \Theta H_1, \tag{7}$$

where Θ is a $d \times l$ matrix of parameters $\theta_{ij} \in \mathbb{K}$. We call the elimination template associated with the matrix H the *parametrized elimination template*.

We note that since the rows of matrix H_1 generate the syzygy module, formula (7) would give us the complete set of solutions to Eq. (6) provided that $\theta_{ij} \in \mathbb{K}[X]$. However, in this paper we restrict ourselves to the much simpler case $\theta_{ij} \in \mathbb{K}$.

In general, the parametrized template may be very large. In the next section we propose several approaches for its reduction.

4. Reduction of the template

4.1. Adjusting parameters by a greedy search

The kth column of matrix H, defined in (7), can be written as $h_k = Z_k c_k$. where Z_k is the kth coefficient matrix whose entries are affine functions in the parameters θ_{ij} and c_k is the related monomial vector. Let $W = \begin{bmatrix} Z_1 & \dots & Z_s \end{bmatrix}$. The columns of matrix W are in one-to-one correspondence with the shifts of polynomials in the expanded system and hence with the rows of the elimination template. Thus, the problem of template reduction leads to the combinatorial optimization problem of adjusting the parameters with the aim of minimizing the number of non-zero columns in W. Below we propose two heuristic strategies for handling this problem. We call the first strategy "row-wise" as it tends to remove the rows of the template. The second strategy is "column-wise" as it removes the columns of W that correspond to excessive monomials and hence to columns of the template.

First, we notice that if a column of matrix W contains an entry, which is a nonzero scalar, then this column can not be zeroed out by adjusting the parameters. Hence, we further assume that all such columns were removed from W.

Row-wise reduction: Let w_k be the kth column of matrix W. To zero out a column of matrix W means to solve linear equations $w_k = 0$. As each row of W has its own set of parameters, solving $w_k = 0$ splits into solving d single equations. For each k, we assign to w_k the score $\sigma(k)$ which is the number of columns that are zeroed out by solving $w_k = 0$. Our row-wise greedy strategy implies that at each step we zero out the column with the maximal score. We proceed while $\sigma(k) > 0$ for at least one k.

Column-wise reduction: Let $\mathcal{E}$ be the set of excessive monomials for the parametrized elimination template. For each $e \in \mathcal{E}$, we denote by $\mathcal{W}_e$ the subset of columns of W such that the respective shifts contain e. For each $e \in \mathcal{E}$, we assign to e the score $\sigma(e)$ which is the number of columns that are zeroed out by solving $w = 0$ for all $w \in \mathcal{W}_e$. Our column-wise greedy strategy implies that at each step we zero out the columns from $\mathcal{W}_e$ corresponding to the excessive monomial e with the maximal score. We proceed while $\sigma(e) > 0$ for at least one e.

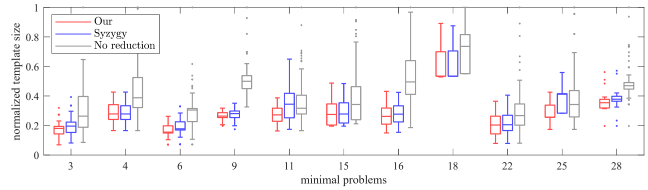


Figure 1. A comparison of our adjusting strategy (Our) with the syzygy-based reduction from [33] (Syzygy). We also show sizes of the initial templates before reduction (No reduction). Each box plot represents the distribution of the normalized template sizes for 100 randomly selected standard monomial bases. The action variable for each basis is also taken randomly. The problem numbering is the same as in Tab. 1 and Tab. 2.

The column-wise strategy is faster as it zeroes out several columns of matrix W at each step. On the other hand, the row-wise strategy outputs smaller templates for some cases. Our automatic template generator tries both strategies and outputs the smallest template.

In Fig. 1, we compare our adjusting strategy with the template reduction method from [33] on several minimal problems. Each box plot on the figure represents the distribution of the normalized template sizes for 100 standard monomial bases corresponding to randomly selected monomial orderings. The action variable for each basis is also taken randomly. The problem instance is the same (fixed) for each problem. For visibility, we also show the sizes of the parametrized templates before applying any reductions.

Our reduction method produces smaller elimination templates in most cases. It can be seen that for some cases the syzygy-based reduction produces templates which are larger than the parametrized templates.

4.2. Schur complement reduction

Proposition 1. *Let M be an elimination template represented in the following block form*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{8}$$

where A is a square invertible matrix and its columns correspond to some excessive monomials. Then the Schur complement of A, i.e., *matrix $M/A = D - CA^{-1}B$, is an elimination template too.*

Proof. See SM Sec. 10. ∎

In practice, Prop. 1 can be used as follows. Suppose that the set of polynomials $F = \{f_1, \dots, f_s\}$ contains a subset, say $F^* = \{f_1, \dots, f_k\}$, such that (i) all polynomials from F^* are sparse, *i.e.*, consist of a relatively small number of terms, and (ii) the coefficients of polynomials from F^* are unchanged for all instances of the problem. Such polynomials may arise, *e.g.*, from the normalization condition. Let an elimination template M for F be represented in the block form (8), where the submatrix $\begin{bmatrix} A & B \end{bmatrix}$ corresponds to the shifts of polynomials from F^*, matrix A is square and invertible, its columns correspond to some excessive monomials and its entries are the same for all instances of the problem. Then, by Prop. 1, we can safely reduce the template by replacing M with the Schur complement M/A.

Since the polynomials from F^* are sparse, the blocks A and B in (8) are sparse too. It follows that the nonzero entries of matrix M/A are simple (polynomial) functions of the entries of M that can be easily precomputed offline. The Schur complement reduction allows one to significantly reduce the template for some minimal problems, see Tab. 1 and Tab. 2 below.

4.3. Removing dependent rows and columns

Proposition 2. *Let M'' be an elimination template of size $s'' \times n''$ whose columns arranged w.r.t. the partition $\mathcal{E} \cup \mathcal{R} \cup \overline{\mathcal{B}}$. Then there exists a template M of size $s \times n$ so that $s \leq s''$, $n \leq n''$ and $n - s = \#\overline{\mathcal{B}}$.*

Proof. See SM Sec. 11. □

By Prop. 2, given an elimination template, say M'', we can always select a maximal subset of linearly independent rows and remove from M'' all the remaining (dependent) rows. The result is an elimination template M'. Similarly, we can always select a maximal subset of linearly independent columns corresponding to the set of excessive monomials and remove from M' all the remaining columns corresponding to the excessive monomials. This is accomplished by twice applying the G–J elimination, first on matrix $M''^\top$ to remove dependent rows and then on the resulting matrix M' to remove dependent columns.

5. Experiments

In this section we test our template generator on two sets of minimal problems. The first one consists of the 21 problems covered in papers [33], [36] and [5]. They provide the state-of-the-art template generators denoted by Syzygy, BeyondGB and SparseR respectively. The results for the first set of problems are presented in Tab. 1.

The second set consists of the 12 additional problems which were not presented in [5, 36]. The results for the second set of problems are reported in Tab. 2. Below we give several remarks regarding Tab. 1 and Tab. 2.

1. The column “std” consists of the smallest templates generated in a standard way using Gröbner bases either from the entire Gröbner fan of the ideal[3] or from 1,000 randomly selected bases in case the Gröbner fan computation cannot be done in a reasonable time. The column “nstd” consists of the smallest templates generated from the 500 quotient space bases found by using the random sampling strategy from [36].

2. The templates marked with $*$ were reduced by the method of Subsect. 4.2. The related minimal problem formulations contain a simple sparse polynomial with (almost) all constant coefficients. For example, the formulations of problems #25 and #26 contain the quaternion normalization constraint $x^2 + y^2 + z^2 + \sigma^2 = 1$, where x, y, z are unknowns and the value of σ is known. All the multiples of this equation can be safely eliminated from the template by constructing the Schur complement of the respective block.

3. The polynomial equations for problem #3 are constructed from the null-space of a 6×9 matrix. We used the sparse basis of the null-space constructed by the G–J elimination as it leads to a smaller elimination template compared to the dense basis constructed by the SVD.

4. The 39×95 elimination template for problem #8 was found w.r.t. the reciprocal of the action variable λ representing the radial distortion, *i.e.*, vector V from (5) was defined as $V = \lambda^{-1} v(\mathcal{B}) - T_{\lambda^{-1}} v(\mathcal{B})$, where the non-standard basis $\mathcal{B}$ consists of monomials that are all divisible by λ. In terms of paper [11], the set $\mathcal{B}$ constitutes the redundant solving basis as it consists of 56 monomials whereas the number of solutions to problem #8 is 52. The four spurious solutions can be filtered out by removing solutions with the worst values of normalized residuals.

5. The initial formulation of problem #15 consists of 5 degree-3 polynomials in 5 variables: 3 rotation parameters and 2 camera center coordinates. As suggested in [5], we first simplified these polynomials using a G–J elimination on the related Macaulay matrix. After that, 2 of 5 polynomials depend only on the rotation variables. The remaining 3 polynomials depend linearly on the camera center variables. We used 2 of these polynomials to solve for the camera center and then substitute the solution into the third polynomial resulting in one additional polynomial of degree 4 in 3 rotation variables only. Hence our formulation of the problem consists of 3 polynomials in 3 variables: 1 polynomial of degree 4 and 2 polynomials of degree 2. It is important to note that (i) the coefficients of the degree-4 polynomial are linearly (and quite easily) expressed in terms of the coefficients of the 3 initial polynomials and (ii) this elimination process does not introduce any spurious roots. We also note that the problem has the following 2-fold symmetry: if x, y, z are the rotation parameters for the Cayley-transform representation, then replacing $x \to y/z$, $y \to -x/z$ and

[3] We used the software package Gfan [21] to compute Gröbner fans.

#	Problem	d	Our		Syzygy [33]	BeyondGB [36]		SparseR [5]
			std	nstd		std	nstd	
1	Rel. pose F+λ 8pt [26]	8	11×19	**7×15**	11×19	11×19	**7×15**	7×16
2	Rel. pose E+f 6pt [6]	9	**11×20**	**11×20**	21×30	**11×20**	**11×20**	**11×20**
3	Rel. pose f+E+f 6pt [52], [29]	15	12×27	**11×26**	31×46	31×46	21×40	12×30
4	Rel. pose E+λ 6pt [26]	26	34×60	**14×40**	34×60	34×60	**14×40**	**14×40**
5	Stitching $f\lambda$+R+$f\lambda$ 3pt [44]	18	48×66	**18×36**	48×66	48×66	**18×36**	**18×36**
6	Abs. pose P4P+fr [7]	16	**52×68**	**52×68**	140×156	54×70	54×70	**52×68**
7	Abs. pose P4P+fr (el. f) [35]	12	**28×40**	**28×40**	**28×40**	**28×40**	**28×40**	**28×40**
8	Rel. pose λ+E+λ 6pt [29]	52	73×125	**39×95**	149×201	–	53×105	**39×95**
9	Rel. pose λ_1+F+λ_2 9pt [29]	24	**76×100**	**76×100**	165×189	87×111	87×111	90×117
10	Rel. pose E+$f\lambda$ 7pt [26]	19	**55×74**	56×75	185×204	69×88	69×88	61×80
11	Rel. pose E+$f\lambda$ 7pt (el. λ) [5]	19	37×56	**22×41**	52×71	37×56	24×43	**22×41**
12	Rel. pose E+$f\lambda$ 7pt (el. $f\lambda$) [30]	19	**51×70**	**51×70**	**51×70**	**51×70**	**51×70**	**51×70**
13	Rolling shutter pose [48]	8	**47×55**	**47×55**	**47×55**	**47×55**	**47×55**	**47×55**
14	Triangulation (sat. im.) [59]	27	**87×114**	**87×114**	88×115	88×115	88×115	**87×114**
15	Abs. pose refractive P5P [19]	16	**57×73**	**57×73**	240×256	112×128	199×215	68×93
16	Abs. pose quivers [25]	20	**65×85**	66×86	169×189	–	68×88	68×92
17	Unsynch. rel. pose [2]	16	159×175	**139×155**	159×175	–	299×315	150×168
18	Optimal PnP (Hesch) [20]	27	**87×114**	**87×114**	88×115	88×115	88×115	**87×114**
19	Optimal PnP (Cayley) [43]	40	**118×158**	**118×158**	**118×158**	**118×158**	**118×158**	**118×158**
20	Optimal pose 2pt v2 [54]	24	**139×163***	141×165*	192×216	–	192×216	176×200
21	Rel. pose E+angle 4pt [37]	20	**99×119***	**99×119***	246×276	–	183×249	–

Table 1. A comparison of the elimination templates of our test minimal problems. We follow the notations from [5, 36] for the problems' names. The columns "std" and "nstd" stand for the templates generated respectively in standard way using Gröbner bases and in non-standard way using heuristics. The minimal templates are shown in bold, the templates which are smaller than the state-of-the-art are shown in blue bold, symbol "–" means a missing template, d is the dimension of the quotient space, $*$: the template is reduced by the method of Subsect. 4.2.

#	Problem	d	Our		Original	Syzygy [33]
			std	nstd		
22	Rel. pose λ+F+λ 8pt [29]	16	**31×47**	**31×47**	32×48	32×48
23	P3.5P+focal [58]	10	**18×28**	19×29	20×43	20×30
24	Gen. P4P+scale [56]	8	**47×55**	**47×55**	48×56	**47×55**
25	Rel. pose E+angle 4pt v2 [41]	20	**16×36***	**16×36***	**16×36***	36×56
26	Gen. rel. pose E+angle 5pt [41]	44	**37×81***	**37×81***	**37×81***	317×361
27	Rel. pose E+fuv+angle 7pt [40]	6	46×52	40×46	$\begin{cases}\mathbf{13\times32}\\ \mathbf{19\times32}\\ \mathbf{11\times20}\\ \mathbf{14\times20}\end{cases}$	66×72
28	Rolling shutter R6P [3]	20	**120×140**	**120×140**	196×216	204×224
29	Opt. pose w dir 4pt [54]	28	**134×162***	144×172*	280×252	203×231
30	Opt. pose w dir 3pt [54]	48	397×445*	**385×433***	1260×1278	544×592
31	L_2 3-view triang. (relaxed) [31]	31	**217×248**	281×312	274×305	231×262
32	Refractive P6P+focal [19]	36	**126×162**	178×214	648×917	636×654
33	Rel. pose $f\lambda$+E+$f\lambda$ 7pt [22]	68	**209×277**	255×323	886×1011	581×659
34	Gen. rel. pose + scale 7pt [24]	140	**144×284**	**144×284**	–	**144×284**

Table 2. A comparison of the elimination templates of our test minimal problems. The columns "std" and "nstd" stand for the templates generated from the standard and non-standard quotient ring bases respectively. We follow the notations from [33] for the problems' names. The minimal templates are shown in bold, the templates which are smaller than the state-of-the-art are shown in blue bold, symbol "–" means a missing template, d is the dimension of the quotient space, $*$: the template is reduced by the method of Subsect. 4.2.

$z \to -1/z$ leaves the polynomial system unchanged. It follows that the problem has no more than 8 "essentially distinct" solutions and hence the template for this problem could be further reduced.

6. Problem #27 was originally solved by applying a cascade of four G–J eliminations to the manually saturated polynomial ideal. We marked the original solver in bold as it is faster than the new single elimination solver (0.4 ms against 0.6 ms).

7. The initial formulation of problem #32 consists of 6 degree-4 polynomials in 6 variables: 3 rotation parameters, 2 camera center coordinates and the focal length. Similarly as we did for problem #15, we first simplified the equations using a G–J elimination on the related Macaulay matrix and then we eliminated the camera center coordinates. This results in 4 equations in 4 unknowns: 1 polynomial of degree

5, 2 of degree 3 and 1 of degree 2. As in the case of problem #15, eliminating variables does not introduce any spurious solutions. We also note that the problem has a 4-fold symmetry meaning that the number of its "essentially distinct" roots is not more than 9. It follows that the template for this problem could be further reduced.

8. The implementation of the new AG, as well as the Matlab solvers for all the minimal problems from Tab. 1 and Tab. 2, are available at http://github.com/martyushev/EliminationTemplates.

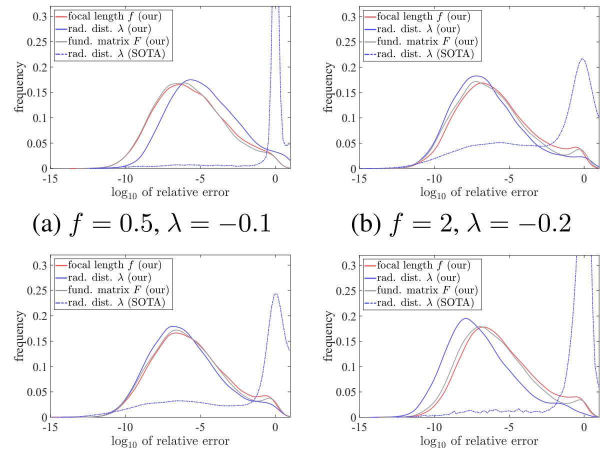
In SM Sec. 13, we test the speed and numerical stability of our solvers.

5.1. Relative pose with unknown focal length and radial distortion

The problem of relative pose estimation of a camera with unknown but fixed focal length and radial distortion can be minimally solved from seven point correspondences in two views. It was first considered in paper [22], where it was formulated as a system of 12 polynomial equations: 1 equation of degree 2, 1 of degree 3, 2 of degree 5, 3 of degree 6 and 5 of degree 7. The 5 unknowns are: the radial distortion parameter λ for the division model from [17], the reciprocal square of the focal length f^{-2} and the thee entries F_{32}, F_{13}, F_{23} of the fundamental matrix F. The related polynomial ideal has degree 68 meaning that the problem generally has 68 solutions.

We started from the same formulation of the problem as in the original paper [22]. We did not manage to construct the Gröbner fan for the related polynomial ideal in a reasonable amount of time (about 24 hours). Instead, we randomly sampled 1,000 weighted monomial orderings so that the respective reduced Gröbner bases are all distinct. We avoided weight vectors where a one entry is much smaller than the others, since the monomial orderings for such weights usually lead to notably larger templates. We also constructed 500 heuristic bases of the quotient ring by using the random sampling strategy from [36]. Then, we used our automatic generator to construct elimination templates for all the bases (both standard and non-standard) and for all the action variables. The smallest template we found this way has size 209×277. It corresponds to the standard basis for the weighed monomial ordering with $f^{-2} > F_{32} > F_{13} > F_{23} > \lambda$ and the weight vector $w = \begin{bmatrix}135 & 81 & 98 & 107 & 68\end{bmatrix}^\top$. The action variable is λ.

The solver from paper [22], based on the elimination template of size 886×1011, is not publically available. However, the results reported in the paper assume that the solver from [22] is much slower than our one (400 ms against 8.5 ms), while the both solvers demonstrate comparable numerical accuracy. The solver based on the 581×659 template generated by the AG from [33] is almost twice slower (about 16 ms) than our solver. Moreover, the



(a) $f = 0.5$, $\lambda = -0.1$ (b) $f = 2$, $\lambda = -0.2$

(c) $f = 1$, $\lambda = -0.5$ (d) $f = 3$, $\lambda = -0.9$

Figure 2. The distribution of relative errors for problem #33 from Tab. 2 on 10^4 trials for different values of focal length f and radial distortion λ. For comparison, we also added the relative error distribution for λ obtained by the state-of-the-art (SOTA) solver from [47].

solver [33] it is unstable and requires additional stability improving techniques, *e.g.*, column pivoting [11]. Hence we compared our solver with the only publicly available state-of-the-art solver from the recent paper [47].

We modeled a scene consisting of seven points viewed by two cameras with unknown but shared focal length f and radial distortion parameter λ. The distance between the first camera center and the scene is 1, the scene dimensions (w×h×d) are 1×1×0.5 and the baseline length is 0.3.

We tested the numerical accuracy of our solver by constructing the distributions of relative errors for the focal length f, radial distortion parameter λ and fundamental matrix F on noise-free image data. We only kept the roots satisfying the following "feasibility" conditions: (i) f^{-2} is real; (ii) $f^{-2} > 0$; (iii) $-1 \leq \lambda \leq 1$. The results for different values of f and λ are shown in Fig. 2.

Our solver failed (*i.e.*, found no feasible solutions) in approximately 2% of trials. The average runtime for the solver from [47] was 2.9 ms which is almost 3 times less than the execution time for our solver (8.5 ms). However, we note that the main parts of the solver from [47] are written in C++, whereas our algorithm is fully implemented in Matlab. This provides a room for further speed up of our solver.

6. Conclusion

We developed a new method for constructing small and stable elimination templates for efficient polynomial system solving of minimal problems. We presented the state-of-the-art templates for many minimal problems with substantial improvement for harder problems.

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** (2017), no. 3, 403–415. 1

[2] Cenek Albl, Zuzana Kukelova, Andrew Fitzgibbon, Jan Heller, Matej Smid, and Tomas Pajdla, *On the two-view geometry of unsynchronized cameras*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4847–4856. 7

[3] Cenek Albl, Zuzana Kukelova, and Tomas Pajdla, *R6p-rolling shutter absolute camera pose*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2292–2300. 7

[4] Daniel Barath and Levente Hajder, *Efficient recovery of essential matrix from two affine correspondences*, IEEE Transactions on Image Processing **27** (2018), no. 11, 5328–5337. 1

[5] Snehal Bhayani, Zuzana Kukelova, and Janne Heikkila, *A sparse resultant based method for efficient minimal solvers*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1770–1779. 1, 2, 6, 7

[6] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla, *3d reconstruction from image collections with a single known focal length*, 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 1803–1810. 2, 7

[7] ______, *New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion*, Asian Conference on Computer Vision, Springer, 2010, pp. 11–24. 2, 7

[8] ______, *Making minimal solvers fast*, 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1506–1513. 2

[9] Martin Byröd, Klas Josephson, and Kalle Åström, *Improving numerical accuracy of Gröbner basis polynomial equation solvers*, 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8. 2

[10] ______, *A column-pivoting based strategy for monomial ordering in numerical Gröbner basis calculations*, European Conference on Computer Vision, Springer, 2008, pp. 130–143. 1, 2, 3

[11] ______, *Fast and stable polynomial equation solving and its application to computer vision*, International Journal of Computer Vision **84** (2009), no. 3, 237–256. 2, 3, 6, 8

[12] David A Cox, John Little, and Donal O'shea, *Using algebraic geometry*, vol. 185, Springer Science & Business Media, 2006. 2, 3, 5

[13] David A. Cox, John Little, and Donald O'Shea, *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*, Springer, 2015. 2, 3

[14] Michel Demazure, *Sur deux problemes de reconstruction*, Ph.D. thesis, INRIA, 1988. 3

[15] 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), –. 2

[16] Martin A Fischler and Robert C Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM **24** (1981), no. 6, 381–395. 1

[17] Andrew W Fitzgibbon, *Simultaneous linear estimation of multiple view geometry and lens distortion*, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2001, pp. 125–132. 8

[18] D. Grayson and M. Stillman, *Macaulay2, a software system for research in algebraic geometry*, 2002, available at http://www.math.uiuc.edu/Macaulay2/. 4

[19] Sebastian Haner and Kalle Åström, *Absolute pose for cameras under flat refractive interfaces*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1428–1436. 7

[20] Joel A Hesch and Stergios I Roumeliotis, *A direct least-squares (DLS) method for PnP*, 2011 International Conference on Computer Vision, IEEE, 2011, pp. 383–390. 7

[21] Anders N. Jensen, *Gfan, a software system for Gröbner fans and tropical varieties*, Available at http://home.imf.au.dk/jensen/software/gfan/gfan.html. 6

[22] Fangyuan Jiang, Yubin Kuang, Jan Erik Solem, and Kalle Åström, *A minimal solution to relative pose with unknown focal length and radial distortion*, Asian Conference on Computer Vision, Springer, 2014, pp. 443–456. 7, 8

[23] Joe Kileel, Zuzana Kukelova, Tomas Pajdla, and Bernd Sturmfels, *Distortion varieties*, Foundations of Computational Mathematics **18** (2018), no. 4, 1043–1071. 2

[24] Laurent Kneip, Chris Sweeney, and Richard Hartley, *The generalized relative pose and scale problem: View-graph fusion via 2d-2d registration*, 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–9. 7

[25] Yubin Kuang and Kalle Åström, *Pose estimation with unknown focal length using points, directions and lines*, Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 529–536. 1, 7

[26] Yubin Kuang, Jan E Solem, Fredrik Kahl, and Kalle Åström, *Minimal solvers for relative pose with a single unknown radial distortion*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 33–40. 7

[27] Yubin Kuang, Yinqiang Zheng, and Kalle Åström, *Partial symmetry in polynomial systems and its applications in computer vision*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 438–445. 2

[28] Zuzana Kukelova, Martin Bujnak, Jan Heller, and Tomáš Pajdla, *Singly-bordered block-diagonal form for minimal problem solvers*, Asian Conference on Computer Vision, Springer, 2014, pp. 488–502. 2

[29] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla, *Automatic generator of minimal problem solvers*, European Conference on Computer Vision, Springer, 2008, pp. 302–315. 1, 2, 3, 7

[30] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla, *A clever elimination strategy for efficient minimal solvers*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4912–4921. 2, 7

[31] Zuzana Kukelova, Tomas Pajdla, and Martin Bujnak, *Fast and stable algebraic solution to L2 three-view triangulation*, 2013 International Conference on 3D Vision-3DV 2013, IEEE, 2013, pp. 326–333. 2, 7

[32] Viktor Larsson and Kalle Åström, *Uncovering symmetries in polynomial systems*, European Conference on Computer Vision, Springer, 2016, pp. 252–267. 2

[33] Viktor Larsson, Kalle Åström, and Magnus Oskarsson, *Efficient solvers for minimal problems by syzygy-based reduction*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 820–829. 1, 2, 4, 5, 6, 7, 8

[34] ———, *Polynomial solvers for saturated ideals*, Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2288–2297. 2

[35] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng, *Making minimal solvers for absolute pose estimation compact and robust*, Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2316–2324. 2, 7

[36] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomas Pajdla, *Beyond Grobner bases: Basis selection for minimal solvers*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3945–3954. 1, 2, 6, 7, 8

[37] Bo Li, Lionel Heng, Gim Hee Lee, and Marc Pollefeys, *A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle*, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 1595–1601. 2, 7

[38] Hongdong Li and Richard Hartley, *Five-point motion estimation made easy*, 18th International Conference on Pattern Recognition (ICPR'06), vol. 1, IEEE, 2006, pp. 630–633. 1

[39] H Christopher Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, Nature **293** (1981), no. 5828, 133–135. 3

[40] Evgeniy Martyushev, *Self-calibration of cameras with euclidean image plane in case of two views and known relative rotation angle*, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 415–429. 7

[41] Evgeniy Martyushev and Bo Li, *Efficient relative pose estimation for cameras and generalized cameras in case of known relative rotation angle*, Journal of Mathematical Imaging and Vision **62** (2020), no. 8, 1076–1086. 7

[42] Teo Mora and Lorenzo Robbiano, *The Gröbner fan of an ideal*, Journal of Symbolic Computation **6** (1988), no. 2-3, 183–208. 2

[43] Gaku Nakano, *Globally optimal DLS method for PnP problem with Cayley parameterization*, Proceedings of the British Machine Vision Conference, 2015, pp. 78.1–78.11. 2, 7

[44] Oleg Naroditsky and Kostas Daniilidis, *Optimizing polynomial solvers for minimal geometry problems*, 2011 International Conference on Computer Vision, IEEE, 2011, pp. 975–982. 2, 7

[45] David Nistér, *An efficient solution to the five-point relative pose problem*, IEEE transactions on pattern analysis and machine intelligence **26** (2004), no. 6, 756–770. 1

[46] David Nistér, Oleg Naroditsky, and James Bergen, *Visual odometry*, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, Ieee, 2004, pp. I–I. 1

[47] Magnus Oskarsson, *Fast solvers for minimal radial distortion relative pose problems*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3668–3677. 8

[48] Olivier Saurer, Marc Pollefeys, and Gim Hee Lee, *A minimal solution to the rolling shutter pose estimation problem*, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 1328–1334. 1, 2, 7

[49] Johannes L Schonberger and Jan-Michael Frahm, *Structure-from-motion revisited*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4104–4113. 1

[50] Noah Snavely, Steven M Seitz, and Richard Szeliski, *Modeling the world from internet photo collections*, International journal of computer vision **80** (2008), no. 2, 189–210. 1

[51] Henrik Stewénius, Christopher Engels, and David Nistér, *Recent developments on direct relative orientation*, ISPRS Journal of Photogrammetry and Remote Sensing **60** (2006), no. 4, 284–294. 1

[52] Henrik Stewénius, David Nistér, Fredrik Kahl, and Frederik Schaffalitzky, *A minimal solution for relative pose with unknown focal length*, Image and Vision Computing **26** (2008), no. 7, 871–877. 7

[53] Bernd Sturmfels, *Solving systems of polynomial equations*, no. 97, American Mathematical Soc., 2002. 3

[54] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson, *City-scale localization for cameras with known vertical direction*, IEEE transactions on pattern analysis and machine intelligence **39** (2016), no. 7, 1455–1461. 7

[55] 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*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7199–7209. 1

[56] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg, *A minimal solution to the generalized pose-and-scale problem*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 422–429. 7

[57] Manuela Wiesinger-Widi, *Gröbner bases and generalized sylvester matrices*, Ph.D. thesis, JKU Linz, 2015. 3

[58] Changchang Wu, *P3.5p: Pose estimation with unknown focal length*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2440–2448. 7

[59] Enliang Zheng, Ke Wang, Enrique Dunn, and Jan-Michael Frahm, *Minimal solvers for 3d geometry from satellite imagery*, Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 738–746. 2, 7