Partial Symmetry in Polynomial Systems and Its Application in Computer Vision

Kuang, Yubin; Zheng, Yinqiang; Åström, Karl

Published in:
[Host publication title missing]

DOI:
10.1109/CVPR.2014.63

2014

Partial Symmetry in Polynomial Systems and its Applications in Computer Vision

Yubin Kuang[1] Yinqiang Zheng[2] Kalle Åström[1]

[1]Centre for Mathematical Sciences
Lund University, Sweden
{yubin,kalle}@maths.lth.se

[2]National Institute of Informatics
Tokyo, Japan
yqzheng@nii.ac.jp

Abstract

Algorithms for solving systems of polynomial equations are key components for solving geometry problems in computer vision. Fast and stable polynomial solvers are essential for numerous applications e.g. minimal problems or finding for all stationary points of certain algebraic errors. Recently, full symmetry in the polynomial systems has been utilized to simplify and speed up state-of-the-art polynomial solvers based on Gröbner basis method. In this paper, we further explore partial symmetry (i.e. where the symmetry lies in a subset of the variables) in the polynomial systems. We develop novel numerical schemes to utilize such partial symmetry. We then demonstrate the advantage of our schemes in several computer vision problems. In both synthetic and real experiments, we show that utilizing partial symmetry allow us to obtain faster and more accurate polynomial solvers than the general solvers[1].

1. Introduction

For many geometric computer vision problems, solving polynomial systems is one of the essential building blocks. For instance, minimal problems can be formulated as polynomial systems with several unknowns e.g. [15]. On the other hand, finding all stationary points of an overdetermined systems [2, 8, 13] has also been formulated as polynomial system. Polynomial systems in computer vision problems are in general of small size (few unknowns) and of low degree. Gröbner basis method has been applied successfully to construct numerical solvers for such systems [17, 11]. Several techniques have been proposed to improve both the stability and the speed for general polynomial solvers [4, 14].

In general, polynomial solvers are problem-specific and structures in the problems can be utilized to further improve the solvers. Recently, in [1], a general technique to exploit full symmetry (i.e. the symmetric pattern is common for all variables) in polynomial systems is proposed to reduce the size of the elimination template as well as improve the stability of the solvers. Utilizing the technique in [1], faster solvers are derived for robust fitting [7] and perspective-n-point problem [19]. While it has been shown that full symmetry exist in several important computer vision problems, it is still a restricted requirement that all variables share the same symmetric pattern. It is of great interest to generalize the technique in [1] for polynomial systems where only a subset of variables share a common symmetric pattern.

Related Works With the introduction of Gröbner basis method for solving polynomial systems in computer vision [17], extensive works have been done in the direction of improving the speed and numerical stability of general numerical solvers. In [4], several basis selection techniques were proposed to enhance numerical stability of the solvers. The work in [11] enables automatic generation of polynomial solver for specific problems and describes a equation removal scheme to speedup polynomial solvers. Other removal strategies have also been studied in [14, 10] The most related work to ours is [1] where full symmetry of the polynomial systems is explored. As [1], our proposed method can be naturally integrated with other techniques for constructing polynomial solvers [4].

Contribution In this paper, we generalize the technique for fully symmetric polynomial systems to partially symmetric systems. This enables a much larger class of problems where we can utilize symmetry to reduce the complexity of the problem and speed up the polynomial solvers. The solution scheme is general and can be readily integrated with existing schemes for improving speed and numerical

[1]This work was supported by the strategic research projects ELLIIT and eSSENCE, Swedish Foundation for Strategic Research projects ENGROSS and VINST (no. RIT08-0043), and the Grants-in-Aid for Scientific Research (no. 25240025) from the Japan Society for the Promotion of Science.

stability. We prove how partial or full symmetry in the solution set is related to the structure of the polynomial equations. This can be used to automatically detect partial or full symmetry. We show that several problems in computer vision exhibit partial symmetry e.g. optimal Euclidean registration and optimal pose from line correspondences. We demonstrate in several experiments that explicitly utilizing such symmetry improve the numerical stability as well as the speed of the polynomial solvers.

2. Symmetry in Polynomial Systems

In this section, we define the concepts for *symmetry* in polynomial systems. To begin with, we consider solving polynomial system as the following problem:

Problem 2.1. *Given a set of m polynomials $f_i(\mathbf{x})$ in n variables $\mathbf{x} = (x_1, \ldots, x_n)$, determine the complete set of solutions to*

$$f_1(\mathbf{x}) = 0, \ldots, f_m(\mathbf{x}) = 0. \tag{1}$$

We denote a monomial $\mathbf{x}^\gamma = x_1^{\gamma_1} x_2^{\gamma_2} \ldots x_n^{\gamma_n}$, where $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ is the vector of exponents. And the degree of $\mathbf{x}^\gamma$ is defined as the sum of exponents $|\gamma| = \gamma_1 + \cdots + \gamma_n$. The set of polynomials f_i's in (1) generate an *ideal* $I = \{l \in \mathbb{C}[\mathbf{x}] : l = \Sigma_i m_i(\mathbf{x}) f_i(\mathbf{x})\}$, where $m_i \in \mathbb{C}[\mathbf{x}]$ are any polynomials and $\mathbb{C}[\mathbf{x}]$ denotes the set of all polynomials in $\mathbf{x}$ over the complex numbers.

In the following, we define partial symmetry in a polynomial and in a solution set, respectively. To facilitate the discussion, for each polynomial $f(\mathbf{x})$ we divide the variables into two sets $\mathbf{x} = \{\mathbf{x}_s, \mathbf{x}_t\}$ such that

$$f(\mathbf{x}) = \sum_k c_k \mathbf{x}_s^{\alpha_k} \mathbf{x}_t^{\beta_k}. \tag{2}$$

Here, c_k's are the coefficients of monomials.

Definition 2.1. *A polynomial $f(\mathbf{x})$ has partial symmetry of type p on a subset of variables $\mathbf{x}_s$ if for each monomial in the polynomial, the sum of the exponents corresponding to $\mathbf{x}_s$ has the same remainder q modulo p.*

A polynomial system is said to have partial symmetry of type p on $\mathbf{x}_s$ if all polynomial equations in the system have symmetry of type p on $\mathbf{x}_s$. Before we discuss partial symmetry in a solution set, we define the *symmetric operator* on $\mathbf{x}_s$ of type p:

$$S^j_{\mathbf{x_s},p}(\mathbf{x}) = (e^{i2\pi j/p}\mathbf{x}_s, \mathbf{x}_t). \tag{3}$$

where $j \in \mathbb{Z}^+$. By definition, we have $S^p_{\mathbf{x_s},p}(\mathbf{x}) = \mathbf{x}$.

Definition 2.2. *A solution set is said to have partial symmetry of type p on a subset of variables $\mathbf{x}_s$ if for each solution $\mathbf{x}^*$, the point $S^j_{\mathbf{x}_s,p}(\mathbf{x}^*)$ is also a solution.*

Example 1. The following polynomial system has partial symmetry ($p = 2$) to x_1, and there is no symmetry for the unknown x_2.

$$\begin{aligned} x_1^2 - x_2^2 - 2 &= 0 \\ x_1^2 - 3x_2 &= 0 \end{aligned} \tag{4}$$

For this system, the two pairs of partial symmetric solutions are $\{[\sqrt{6} \;\; 2]^T, [-\sqrt{6} \;\; 2]^T\}$ and $\{[\sqrt{3} \;\; 1]\}^T, [-\sqrt{3} \;\; 1]^T\}$.

The following theorem shows that partial symmetry in a polynomial system has a solution set that is partially symmetric. On the other hand, the existence of a partially symmetric solution set implies that the polynomial system is partially symmetric.

Theorem 2.1. *A system of polynomial equations, where every polynomial has partial symmetry of type p on the subset of variables $\mathbf{x}_s$ has a solution set with partial symmetry of type p on $\mathbf{x}_s$. Vice versa, each system of polynomial equations, whose solution set has partial symmetry of type p on $\mathbf{x}_s$, can be written as a set of polynomial equations with partial symmetry of type p on $\mathbf{x}_s$.*

Proof. We first prove that partial symmetry in polynomial systems indicates partially symmetric solution sets. Assume for a polynomial equation $f(\mathbf{x}) = 0$ in the system, the sum of the exponents $|\gamma_k|$ has constant remainder q modulo p i.e. $\forall k$, $q \equiv |\gamma_k| \bmod p$. Let $b = e^{i2\pi/p}$, we have $b^{|\gamma_k|} = e^{i2\pi|\gamma_k|/p} = b^q$. Then for $f\big(S^1_{\mathbf{x}_s,p}(\mathbf{x})\big) = f\big((b\mathbf{x}_s, \mathbf{x}_t)\big)$,

$$\begin{aligned} f\big((b\mathbf{x}_s, \mathbf{x}_t)\big) &= \sum_k c_k (b\mathbf{x}_s)^{\gamma_k} \mathbf{x}_t^{\beta_k} = \sum_k b^{|\gamma_k|} c_k \mathbf{x}_s^{\gamma_k} \mathbf{x}_t^{\beta_k} \\ &= \sum_k b^q c_k \mathbf{x}_s^{\gamma_k} \mathbf{x}_t^{\beta_k} = b^q \sum_k c_k \mathbf{x}_s^{\gamma_k} \mathbf{x}_t^{\beta_k} \\ &= b^q f(\mathbf{x}). \end{aligned} \tag{5}$$

Thus if $f(\mathbf{x}^*) = 0$, then $f\big((b\mathbf{x}_s^*, \mathbf{x}_t^*)\big) = b^q f(\mathbf{x}^*) = 0$. One can prove the same for $f\big(S^j_{\mathbf{x}_s,p}(\mathbf{x}^*)\big) = f\big((b^j\mathbf{x}_s, \mathbf{x}_t)\big)$ by induction. This proves the assertion in one direction.

We then prove the existence of partially symmetric solutions indicates partial symmetry in the corresponding polynomial systems. Assume for a certain p that for every solution $\mathbf{x}^* = (\mathbf{x}_s^*, \mathbf{x}_t^*)$ holds also

$$f\big((b^j\mathbf{x}_s^*, \mathbf{x}_t^*)\big) = 0, \quad j = 0, \ldots, p-1. \tag{6}$$

Divide the polynomial into p parts according to $|\gamma_\mathbf{k}| \bmod p$ so that

$$f(\mathbf{x}) = g_0(\mathbf{x}) + g_1(\mathbf{x}) + \ldots + g_{p-1}(\mathbf{x}). \tag{7}$$

Then we have

$$\begin{aligned} f\big((\mathbf{x}_s, \mathbf{x}_t)\big) &= g_0(\mathbf{x}) + g_1(\mathbf{x}) + \ldots + g_{p-1}(\mathbf{x}), \\ f\big((b\mathbf{x}_s, \mathbf{x}_t)\big) &= g_0(\mathbf{x}) + bg_1(\mathbf{x}) + \ldots + b^{p-1}g_{p-1}(\mathbf{x}), \\ \vdots \qquad & \quad \vdots \qquad \quad \vdots \\ f\big((b^{p-1}\mathbf{x}_s, \mathbf{x}_t)\big) &= g_0(\mathbf{x}) + b^{p-1}g_1(\mathbf{x}) + \ldots + bg_{p-1}(\mathbf{x}). \end{aligned}$$

which is equivalent to the following linear system

$$\mathbf{F}(\mathbf{x}) = \mathbf{H}_p \mathbf{G}(\mathbf{x}) \tag{8}$$

where

$$\begin{aligned} \mathbf{F}(\mathbf{x}) &= \begin{bmatrix} f\left((\mathbf{x}_s, \mathbf{x}_t)\right) & \dots & f\left((b^{p-1}\mathbf{x}_s, \mathbf{x}_t)\right) \end{bmatrix}^T, \\ \mathbf{G}(\mathbf{x}) &= \begin{bmatrix} g_0(\mathbf{x}) & g_1(\mathbf{x}) & \dots & g_{p-1}(\mathbf{x}) \end{bmatrix}^T \end{aligned}$$

and

$$\mathbf{H}_p = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & b & b^2 & \dots & b^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & b^{p-1} & b^{p-2} & \dots & b \end{bmatrix}. \tag{9}$$

From (6), it can be seen that $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$. Since $\mathbf{H}_p$ is invertible for all p (it is basically the matrix representing the discrete Fourier transform of p-vectors), it follows that $g_j(\mathbf{x}^*) = 0$, for all $j = 0, \dots, p-1$. Thus if there exist a set of partially symmetric solutions, then it follows that each polynomial $f_i(\mathbf{x})$ can be split into p parts i.e. $g_{ij}(\mathbf{x})$, where each part has $\mathbf{x}^*$ as the solution. □

From the definition of partial symmetry of a polynomial system, we can easily interpret that *full symmetry* (symmetry in all variables) introduced in [1] is a special case of partial symmetry. Specifically, it corresponds to the cases where $\mathbf{x}_s = \mathbf{x}$. Thus, by Theorem 2.1, we have generalized the full symmetry to partial symmetry.

3. Revisit Gröbner Basis Methods

In this section, we first describe the action matrix method for solving polynomial systems. We then present briefly the polynomial solving technique in [4]. These two techniques serve as the basic building blocks for our symmetric polynomial solvers.

3.1. Action Matrix Method

In this paper, we consider polynomial systems where the zero set V of (1) (or the ideal I) is finite i.e. V is a point set. If V is a finite point set, it can be proved that the quotient space $\mathbb{C}[\mathbf{x}]/I$ is also finite-dimensional [5]. Furthermore, if I is radical (I is equal to the complete set of polynomials vanishing on V), then one can show that $\mathbb{C}[\mathbf{x}]/I$ is isomorphic to $\mathbb{C}^r$, where $r = |V|$ is the number of solutions to the polynomial system [5].

The action matrix method is an multivariate extension of companion matrix for solving uni-variate polynomial equations. To start with, we consider first the linear mapping $T_{a(\mathbf{x})} : f(\mathbf{x}) \mapsto a(\mathbf{x})f(\mathbf{x})$ in the r-dimensional $\mathbb{C}[\mathbf{x}]/I$ where $a(\mathbf{x}) \in \mathbb{C}(\mathbf{x})$. In this paper, we limit the choice of $a(\mathbf{x})$ to be a monomial instead of any polynomial and we call $a(\mathbf{x})$ as *action monomial*. While in general $a(\mathbf{x})$ is chosen as one of the variables, we will discuss in later section that choosing $a(\mathbf{x})$ is one of key steps in utilizing partial symmetry. Given that $\mathbb{C}(\mathbf{x})/I$ is finite dimensional, one can choose a linear basis of monomials $\mathcal{B} = \{\mathbf{x}^{\alpha_1}, \dots, \mathbf{x}^{\alpha_r}\}$ for $\mathbb{C}(\mathbf{x})/I$. Now the mapping $T_{a_{(\mathbf{x})}}$ can be represented as a $r \times r$ matrix $\mathbf{m}_a$, which is the so-called *action matrix*. The solutions of the polynomial system is closely related to this matrix. The eigenvalues of m_a are the values of $a(\mathbf{x})$ evaluated at the solution points i.e. V. On the other hand, the eigenvectors of the m_a^T are the values of the basis monomials in $\mathcal{B}$ evaluated at the solution points.

3.2. Constructing Action Matrix

In this section, we review several important techniques for constructing action matrix in a numerically stable manner including the single eliminate scheme and the basis selection technique.

The single elimination technique has been widely adapted e.g. [4, 11]. It starts by multiplying the equations in (1) by a set of multiplication monomials and produces an equivalent and expanded set of equations. This is in contrast to the Buchberger's algorithm for computing Gröbner bases where equations are generated incrementally. By stacking the coefficients of the expanded set of equations in a coefficient matrix $\mathbf{C}_{\text{exp}}$ which is usually called *elimination template*, we have

$$\mathbf{C}_{\text{exp}} \mathbf{X}_{\text{exp}} = 0. \tag{10}$$

where $\mathbf{X}_{\text{exp}}$ is a vector of the set of monomials in the expanded equations. In general, the set of multiplication monomials can be chosen such that the resulting equations are all up to a certain degree.

To select $\mathcal{B}$ in a numerically stable way, we have used the column-pivoting scheme for basis selection [3]. To enable basis selection, one first partition the set of all monomials $\mathcal{M}$ occurring in the expanded set of equations as $\mathcal{M} = \mathcal{E} \bigcup \mathcal{R} \bigcup \mathcal{P}$. Specifically, $\mathcal{P}$ (permissible monomials) represent the set of monomials that remain in $\mathcal{M}$ after multiplying with $a(\mathbf{x})$. The reducible set $a(\mathbf{x})\mathbf{x}^{\alpha_k} \notin \mathcal{P}$ for $\mathbf{x}^{\alpha_k} \in \mathcal{P}$ is denoted as $\mathcal{R}$. We denote the remaining monomials as the excessive set $\mathcal{E}$. By reordering the monomials such that $\mathcal{E} > \mathcal{R} > \mathcal{P}$, we yield

$$\begin{bmatrix} \mathbf{C}_{\mathcal{E}} & \mathbf{C}_{\mathcal{R}} & \mathbf{C}_{\mathcal{P}} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}} \end{bmatrix} = 0. \tag{11}$$

The key idea of of [3] is to select $\mathcal{B}$ adaptively from a permissible set $\mathcal{P}$ where $|\mathcal{P}| > r$. The first step eliminate the monomials in $\mathcal{E}$:

$$\begin{bmatrix} \mathbf{U}_{\mathcal{E}_1} & \mathbf{C}_{\mathcal{R}_1} & \mathbf{C}_{\mathcal{P}_1} \\ 0 & \mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_2} \\ 0 & 0 & \mathbf{C}_{\mathcal{P}_3} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}} \end{bmatrix} = 0, \tag{12}$$

where $\mathbf{U}_{\mathcal{E}_1}$ and $\mathbf{U}_{\mathcal{R}_2}$ are upper triangular. One can remove the top rows of the coefficient matrix involving the $\mathcal{E}$

$$\begin{bmatrix}\mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_2} \\ 0 & \mathbf{C}_{\mathcal{P}_3}\end{bmatrix}\begin{bmatrix}\mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}}\end{bmatrix} = 0. \tag{13}$$

In the second elimination step, the goal is to reduce $\mathbf{C}_{\mathcal{P}_3}$ into upper triangular matrix. In [3], column-pivoting QR is utilized to improve the stability, which introduces a permutation $\mathbf{C}_{\mathcal{P}_3}\Pi$, where Π is a permutation matrix. The basis is selected as the last r monomials after the reordering Π i.e. $\begin{bmatrix}\mathbf{X}_{\mathcal{P}'} & \mathbf{X}_{\mathcal{B}}\end{bmatrix}^T$. This gives

$$\begin{bmatrix}\mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_4} & \mathbf{C}_{\mathcal{B}_1} \\ \mathbf{0} & \mathbf{U}_{\mathcal{P}_3} & \mathbf{C}_{\mathcal{B}_2}\end{bmatrix}\begin{bmatrix}\mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}'} \\ \mathbf{X}_{\mathcal{B}}\end{bmatrix} = 0. \tag{14}$$

To this end, monomials in $\mathcal{R}$ and $\mathcal{P}'$ are linear combinations of monomials in $\mathcal{B}$:

$$\begin{bmatrix}\mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}'}\end{bmatrix} = -\begin{bmatrix}\mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_4} \\ \mathbf{0} & \mathbf{U}_{\mathcal{P}_3}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{C}_{\mathcal{B}_1} \\ \mathbf{C}_{\mathcal{B}_2}\end{bmatrix}\mathbf{X}_{\mathcal{B}}. \tag{15}$$

By finding the corresponding indices of the monomials $\{a(\mathbf{x})\mathbf{x}^{\alpha_k} \mid \forall \mathbf{x}^{\alpha_i} \in \mathcal{B}\}$ in $[\mathbf{X}_{\mathcal{R}} \quad \mathbf{X}_{\mathcal{P}'}]$, the action matrix can be extracted from the linear mapping (15).

4. Utilizing Partial Symmetry

In this section, we present methods to integrate partial symmetry into the action matrix method.

4.1. Partially Symmetric Action Matrix

Recall from Theorem 2.1 that for a type-p partially symmetric polynomial system in $\mathbf{x}_s$, there exist a set of type-p partially symmetric solutions in the form of $(e^{i2\pi j/p}\mathbf{x}_s, \mathbf{x}_t)$ where $j = 0, \dots, p-1$. This suggests that there is a p-fold ambiguity in the r solutions, which can be utilized to simplify the action matrix construction step. To simplify the discussion, we assume that there is no zero solution or one have used the scheme in [1] to remove the zero solution. The idea is to construct a linear mapping $T_{a(\mathbf{x})} : f(\mathbf{x}) \mapsto a(\mathbf{x})f(\mathbf{x})$ that preserves the underlying partial symmetry of the system.

To achieve this, we first need to choose the action monomial $a(\mathbf{x})$ such that the sum of exponents of $\mathbf{x}_s$ in $a(\mathbf{x})$ is p. This follows that the p ambiguous solutions collapse into a single solution point in $a(\mathbf{x})$ which effectively reduces the dimension of the solution space to $r_p = \frac{r}{p}$ [2]. Thus, instead of considering the original solution space, we can express the reduced space with a monomial basis $\mathcal{B}_p$ of size r_p. The advantage of the reduced action matrix is obvious in that it not only simplify the last eigenvalue decomposition step. We show in the experimental section that it also leads a elimination template of smaller size which speedups the solver even more.

The next step is to construct a elimination template with partially symmetry in mind. The idea again is to generate the expanded set of equations that preserve the symmetry in $\mathbf{x}_s$. To facilitate the elimination step for partially symmetric systems, we propose a scheme to achieve this: choose the multiplication monomials that results in the same remainder modulo p across the set of different equations (not only within each equation). This scheme ensures good overlapping of the expanded monomials between different equations, which improves the efficiency and stability of the elimination step. We will illustrate these schemes in Section 5.3 with detailed examples.

4.2. Extracting Solutions

Once we have constructed the reduced action matrix $\mathbf{m}_{r_p}$, we can extract the solutions from the eigenvector $\mathbf{v}$'s of $\mathbf{m}_{r_p}^T$. From the action matrix method, we know that the eigenvectors v are values of the basis monomials $\mathcal{B}_p$ evaluated at the solutions up to unknown scales. Specifically, we have for each element in $\mathbf{v}$

$$\lambda \mathbf{v}_k = x_1^{\alpha_{k1}} x_2^{\alpha_{k2}} \dots x_n^{\alpha_{kn}}$$

where $k = 1, \dots, r_p$ and λ is a unknown constant. In general polynomial solvers where symmetry is not explored, the simplest scenario is that all the first-order monomials as well as the constant term i.e. $\{x_1, \dots, x_n, 1\}$ are in $\mathcal{B}$. In this case, the solutions can be extracted by reading off the corresponding values of $\{\lambda x_1, \dots, \lambda x_n, \lambda\}$ from $\mathbf{v}$ and the solutions for $\{x_1, \dots, x_n\}$ can be calculated via division by λ. In fact, we can also extract the solutions directly if the first-order monomials are in $\mathcal{R}$ or $\mathcal{P}'$, which can be written as linear combination of $\mathcal{B}$ based on (15). However, for type-p partially symmetric cases where $p > 1$, the general idea is to find a mapping from monomials in $\mathcal{B}_r$ to x_i^p for $i = 1, \dots, n$ and the p-fold ambiguity of x_i's can be solved directly. For example, if we know $v_1 = \lambda x_1 x_2$ and $v_2 = \lambda x_1 x_2^3$ are in the basis, one can calculate $x_2^2 = v_2/v_1$. In general, this mapping is not unique. To find one of such mappings automatically, a general two-step scheme involving (i) a random sampling step and (ii) solving an integer linear system was derived in [1]. This scheme can be generalized directly without any modification to partially symmetric systems. However, it has been seen in our experiments that the stability of the mappings found by such a general scheme vary and most of them can be very unstable. This is due to the fact that most of these mappings involve (i) evaluation of the solutions for monomials of high degrees (ii) numerical operations e.g. division of monomial of high degrees. Therefore, we have derived proper but numerically stable mapping for specific problems. The general guide-

[2] r is divisible by p, if there exists no zero solution.

line for choose such mapping is to avoid monomials of high degrees which in general introduces ill-condition divisions.

4.3. Detecting Partial Symmetry

Based on the Theorem 2.1, we describe a simple strategy for detecting partially symmetric polynomial system. It is an exhaustive scheme and involves combinatorial search over all subsets of variables. However, for general problems in computer vision where the number of variables are among 2 to 10, the search is completely feasible. If d is lowest degree of all the polynomials $f_i(\mathbf{x})$'s, we check for each subset of $\mathbf{x}$, whether type-p ($2 \leq p \leq d$) partial symmetry is fulfilled (by checking the remainder of sum of exponents of the corresponding subset modulo p for each of the polynomial in the system.

5. Applications

In this section, we discuss partial symmetry in the context of geometric problems in computer vision. We show that formulating these problem in a straightforward manner yields polynomial systems with partial symmetry. While the formulations are straightforward as well as avoid certain degeneracy, the resulting polynomial systems have not been solved before due to their difficulty. We illustrate the proposed techniques in details for these examples and obtain faster and more stable solvers than previous state-of-the-art general polynomial solvers.

5.1. Optimal Euclidean Registration

First we study the Euclidean registration problem given point-point, point-line or point-plane correspondences [16]. This problem has been solved using a branch-and-bound method that exploits the quasi-convexity property of the minimization problem. For simplicity, we use the point-plane correspondences only.

Problem 5.1. *(Optimal Euclidean Registration) Given n points x_i, and their corresponding planes in another coordinate system, each of which is represented by the normal e_i and a supporting point y_i, to find the optimal rotation $\mathbf{R}^*$ and translation $\mathbf{t}^*$ such that the*

$$\{\mathbf{R}^*, \mathbf{t}^*\} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{n} \left(e_i^T(\mathbf{R}x_i + \mathbf{t} - y_i)\right)^2 \tag{16}$$

Given $\mathbf{R}$, the translation $\mathbf{t}$ can be directly solved as

$$\mathbf{t} = \left(\sum_{i=1}^{n} e_i e_i^T\right)^{-1} \sum_{i=1}^{n} e_i e_i^T \left(y_i - \mathbf{R}x_i\right). \tag{17}$$

After parameterizing $\mathbf{R}$ by the unit quaternion $\mathbf{q}$ and plugging $\mathbf{t}$ back into (16), we obtain a constraint optimization problem

$$\begin{aligned} \{\mathbf{q}^*\} &= \arg\min_{\mathbf{q}} \sum_{i=1}^{n} \left(e_i^T(\mathbf{R}(\mathbf{q})x_i + \mathbf{t}(\mathbf{q}) - y_i)\right)^2, \\ &s.t. \ \|\mathbf{q}\|_2^2 = 1. \end{aligned} \tag{18}$$

5.2. PnL Problem

The Perspective-n-Line (PnL) problem is to estimate the absolute pose of a calibrated camera by using n known lines and their image projections. It was studied in [13].

Problem 5.2. *(Perspective-n-Line) Given n lines with direction l_i in the world framework, and their corresponding image lines, each of which determines a plane passing through the optical center with normal e_i, to find the optimal rotation $\mathbf{R}^*$ such that*

$$\{\mathbf{R}^*\} = \arg\min_{\mathbf{R}} \sum_{i=1}^{n} \left(e_i^T \mathbf{R} l_i\right)^2. \tag{19}$$

After the rotation $\mathbf{R}$ is determined, the estimation of translation $\mathbf{t}$ becomes trivial.

To parameterize $\mathbf{R}$ by the unit quaternion $\mathbf{q}$ would lead to a constraint optimization problem

$$\begin{aligned} \{\mathbf{q}^*\} &= \arg\min_{\mathbf{q}} \sum_{i=1}^{n} \left(e_i^T \mathbf{R}(\mathbf{q}) l_i\right)^2, \\ &s.t. \ \|\mathbf{q}\|_2^2 = 1. \end{aligned} \tag{20}$$

5.3. Constructing Polynomial Solvers

To solve for global optimal of the geometric or algebraic errors, we use the first order optimality condition i.e. to find all stationary points of the error function. To do that, we calculate the partial derivative of the functions with respect to the unknowns in the quaternion $\mathbf{q} = \{a, b, c, d\}$ as well as the Lagrange multiplier w. Both problems yield a mixture of cubic and quadratic equations in 5 unknowns. Specifically, for optimal Euclidean registration, we have a polynomial system in the following form:

$$\begin{bmatrix} \mathbf{V} & \mathbf{0}_{4\times4} & -\mathbf{I} & \mathbf{0}_{4\times1} \\ \mathbf{0}_{1\times24} & \mathbb{1}_{1\times4} & \mathbf{0}_{1\times4} & -1 \end{bmatrix} \mathbf{x} = \mathbf{0} \tag{21}$$

where $\mathbf{V}$ is a 4×24 coefficient matrix calculated from each specific problem and $\mathbf{x}$ is the monomial vector $[a^3, a^2b, a^2c, a^2d, ab^2, abc, abd, ac^2, acd, ad^2, b^3, b^2c, b^2d, bc^2, bcd, bd^2, c^3, c^2d, cd^2, d^3, a, b, c, d, a^2, b^2, c^2, d^2, wa, wb, wc, wd, 1]^T$.

As for the PnL problem, we have the following polynomial system:

$$\begin{bmatrix} \mathbf{U} & \mathbf{0}_{4\times4} & -\mathbf{I} & \mathbf{0}_{4\times1} \\ \mathbf{0}_{1\times20} & \mathbb{1}_{1\times4} & \mathbf{0}_{1\times4} & -1 \end{bmatrix} \mathbf{x} = \mathbf{0} \tag{22}$$

where $\mathbf{U}$ is a 4×20 coefficient matrix for each specific problem and $\mathbf{x}$ is the monomial vector $[a^3, a^2b, a^2c, a^2d, ab^2, abc, abd, ac^2, acd, ad^2, b^3, b^2c, b^2d, bc^2, bcd, bd^2, c^3, c^2d, cd^2, d^3, a^2, b^2, c^2, d^2, wa, wb, wc, wd, 1]^T$.

By checking the polynomials in both systems, we can see that the variables in the subset $\mathbf{x}_s = \{a, b, c, d\}$, only appear with 3^{rd} and 1^{st} degree in the first 4 equations ($p = 2, q = 1$), and only 2^{nd} and 0^{th} degree in the last equation ($p = 2, q = 0$). Thus, these two polynomial systems are both partially symmetric of type 2 to $\{a, b, c, d\}$ according to Theorem 2.1. Note that, these partial symmetries can not be resolved via variable substitution.

By using tools in algebraic geometry [6], we verify that there are in general 80 solutions to these two polynomial systems or equivalently 40 pairs of partially symmetric solutions. Using automatic generator in [11] which does not utilize partial symmetry, we obtain general solvers that solve for the 80 solutions directly. The elimination templates of these solver are of size 1523×1603 and 688×788 for the Euclidean registration and the PnL, respectively. Note that in this elimination template is obtained after the build-in optimization for template size [11]. There is little possibility to reduce the size of this template further.

Now we discuss in details the construction of our partially symmetric solvers. It turns out the elimination template and solution extraction scheme work for both problems due to their similarity in structures. We can choose any quadratic monomials in $\{a, b, c, d\}$ as the action monomial to utilize type-2 partially symmetry. There is no significant effect for the choice of action monomial on the numerical stability for the two problems. Here we use a^2 for the following discussion. The second step is to choose the set of multiplication monomials. The idea is to ensure the expanded set of monomials coincide between different polynomials. This will facilitate the elimination step so that the numerical stability is improved. To start with, for the first 4 equations, we choose $\{\mathcal{H}_1, \mathcal{H}_3, a^2\mathcal{H}_3, w\mathcal{H}_1, w\mathcal{H}_1, wa^2\mathcal{H}_3, w^2\mathcal{H}_1, w^2\mathcal{H}_1, w^2a^2\mathcal{H}_3\}$ as the set of multiplication monomials. Here $\mathcal{H}_k$ denotes the set of monomials in $\{a, b, c, d\}$ where the sum of exponents is k. In the resulting expanded set, the sum of exponents for variables $\{a, b, c, d\}$ in the monomials are all even. Correspondingly, the multiplication monomials for the last equation are chosen as $\{\mathcal{H}_2, \mathcal{H}_4, a^2\mathcal{H}_4, w\mathcal{H}_2, w\mathcal{H}_4, wa^2\mathcal{H}_4, w^2\mathcal{H}_2, w^2\mathcal{H}_4, w^2a^2\mathcal{H}_4, w, w^3\}$. This also yields a set of expanded monomials where the sum of exponents for variables $\{a, b, c, d\}$ are even. This results in a stable elimination template is of size 770×854, which is already much smaller than the 1523×1603 elimination template for the Euclidean registration problem. With further tuning with similar equation removal technique in [11], we obtain a more compact elimination template of size 433×487. This is used for all our experiments later. With this elimination template, we follow the basis selection technique where we have chosen the permissible set as the last 60 monomials (in *grevlex* order) and construct the 40×40 action matrix.

The last remaining step is to extract solutions by utilizing eigenvectors of the transpose of the action matrix. To enhance the numerical stability, we have derived the following extraction scheme. The first observation is that, for these two problems, one can enforce the constant term i.e. $\mathbf{x}^0$ to be in $\mathcal{B}, \mathcal{R}$ or $\mathcal{P}'$ without breaking the type-2 partially symmetry. Extracting the values for the unknown constant λ's is simply reading off the corresponding values from the vectors. We note also that $\{\lambda a^2, \lambda ac, \lambda ab, \lambda ad\}$ are expressible by linear combination of the basis, which means that one can obtain their values up to an common unknown constant for each solution i.e. $\{\lambda a^2, \lambda ac, \lambda ab, \lambda ad\}$. Thereafter, the solutions of a^2 can be calculated by division with the values of the constant term. We retrieve the two sets of solutions for a by taking square root. The values for $\{b, c, d\}$ can be extracted using similar division given that λ and a are known. The 2-fold ambiguity of the solutions are handled naturally in the extraction step which is much simpler than solving the directly as in the general methods. In case of near-degenerated configuration where $a \approx 0$, the division when we extract $\{b, c, d\}$ is ill-conditioned. In those case, we can extract the solutions with $\{b^2, b^4, ba, bc, bd\}$ etc. in a similar way to avoid such degeneracy. Note that these extraction steps are fast given that the bottleneck is generally in the elimination step. Therefore, we can extract all possible solutions in an very efficient way to avoid degeneracy. This is superior to the schemes in [8] which requires solve several different polynomial problems with specific solvers.

6. Other Examples

Besides the two problems presented above, partial symmetry also exists in other geometric problems in computer vision. For instance, any problems that involve Cayley's parameterization can be reformulated with the unit-norm constraint on the quaternion. These problems generally are fully symmetric or partially symmetric of type-2 to the quaternion e.g. [9]. On the other hand, estimating fundamental matrix with radial distortion in [12] is partially symmetric of type-2 to $\{f_{1,1}, ..., f_{3,3}\}$ if we fix the scale of the fundamental matrix by enforcing $\|f\|_2 = 1$ instead of setting $f_{3,3}$ to 1. The same holds for other related problems that fix the scale in a similar way.

7. Experiments

Optimal Euclidean Registration For this experiments, we simulate point-to-plane correspondence in 3D randomly. We first study the numerical stability of the general solver generated by [11] and our solver that utilizes partial symme-

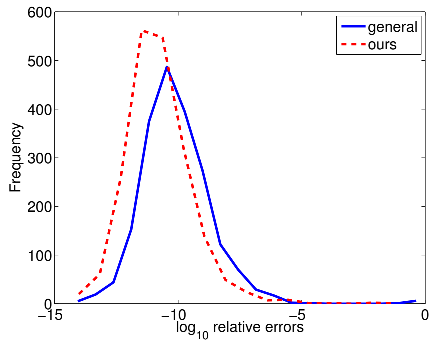

Figure 1. Numerical stability of the general polynomial solver [11] and our symmetric solver for the optimal Euclidean registration problem. The histogram of log_{10} relative errors of $\{a, b, c, d\}$ for 2000 noise-free random problems is shown.

try. From Figure 1, we can see that our method is superior to the general solver with respect to the stability. On the other hand, our solver is around three times faster than the general solver (Table 1). The formulation in [16] is a quadratic programming problem and a branch-and-bound scheme (bnb) for the global optima was derived. It is a much more difficult problem to solve and is in general very slow with increasing number of points. Our formulation along with our solver is much fast and guarantee to find the global optimal of the same geometric error.

bnb* [16]	General [11]	Ours
> 1s	75ms	23ms

Table 1. Average time performance of different solvers for the optimal Euclidean registration problem. ($*$) based on the time reported in [16]. The other timing are measured on a MacBook Air with 1.8 Ghz i5 CPU.

PnL problem For the synthetic experiments in this section, we first randomly generate 3D lines at around the origin, with cameras pointing towards the origin approximately. Then we calculate 2D projections of the lines onto the image plane. We perturb endpoints of lines to simulate noise. We will first look at the numerically stability of the proposed solvers under different configurations. Here by *Cayley degeneracy* for the camera pose, we mean one or several of the variables in $\{a, b, c, d\}$ is equal to (Cayley-degnerate) or close to 0 (near-Cayley-degenerate). For methods in [13, 8] where one of the $\{a, b, c, d\}$ is assumed to be 1, we can see that they degenerate in cases when that specific variable is actually 0. In Figure 2, we first observe that our partially symmetric solver is better than the general solvers across different experiments. Our solver is faster (23ms) compared to the general solver (55ms) which has a larger elimination template. On the other hand, we

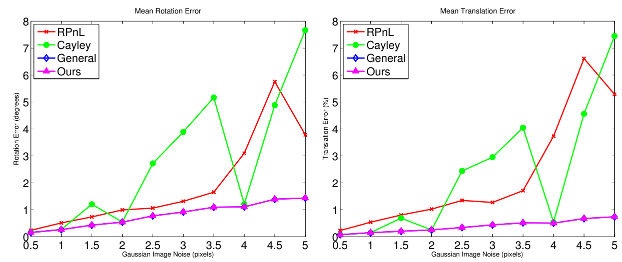

Figure 3. Noise sensitivity of different PnL solvers with varying noise levels for fixed n = 10 lines.

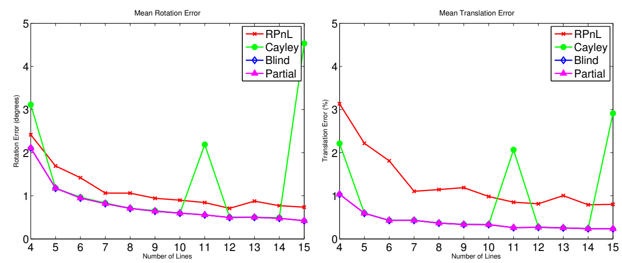

Figure 4. Noise sensitivity of different PnL solvers with varying number of lines with fixed noise level (2 pixels).

can see the solver based on Cayley's parameterization of the quaternion [13] performs better than both our solver and the general solver for random configurations. However, when the configuration is close to degeneracy, the performance of such solve deteriorate drastically (Figure 2, mid). And for degenerated cases (e.g. $a = 0$), the solvers fail completely. While one can argue that running different solvers for different degeneracy mitigates these issues, but it undermines the intrinsic structure of the problem.

We then study the performance of the solvers under noise. In this experiment, we generate camera poses in a fully-random manner. With the formulation in this paper, we can see that the solvers are more robust to degeneracy which causes the large variance in mean errors for Cayley-based method. The RPnL method described in [18] yields better results than Cayley-based method, but still inferior to the formulation here. With the same formulation, our solver performs similarly to the general solver under varying noise levels and number of lines, while being much faster.

8. Conclusion

We present a general framework for utilizing partial symmetry in solving polynomial systems. We study and prove the correspondence between partially symmetric polynomials and solution sets that are partially symmetric. We have also identified two example problems in computer vision that have partial symmetry. We verify the improvements gained by utilizing partially symmetry in both speed and

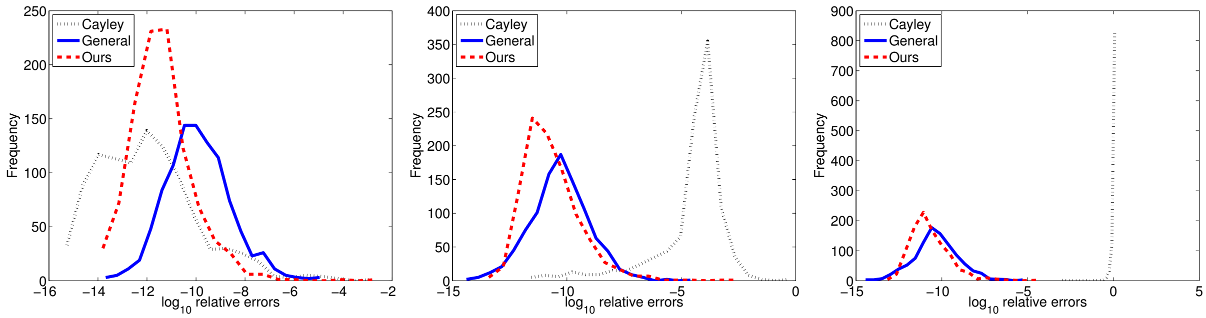

Figure 2. Numerical stability of the Cayley PnL solver [13], general polynomial solver [11] and our symmetric solver under varying configurations (from left to right, 'fully random', 'near-Cayley degeneracy', 'Cayley degeneracy'). The histogram of log_{10} relative errors 2000 noise-free random problems is shown.

accuracy in these problems. The solvers derived using our methods are both faster and more stable than previous general solvers.

As future work, it is of practical importance to achieve automatic detection and reformation of partially symmetric polynomial system. While the techniques presented in this paper can be combined with previous optimization schemes for polynomial solvers, it is of interest to see whether specific optimization scheme can be derived for partially symmetry systems. Moreover, it is important to derive schemes for automatically selecting mapping for solution extraction in a numerically stable manner. On the other hand, the exploration and integration of other types of symmetry in solving polynomial system are of particular interest.

References

[1] E. Ask, Y. Kuang, and K. Åström. Exploiting p-fold symmetries for faster polynomial equation solving. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3232–3235. IEEE, 2012. 1, 3, 4

[2] M. Byröd, K. Josephson, and K. Åström. Fast optimal three view triangulation. In *Asian Conference on Computer Vision*, 2007. 1

[3] M. Byröd, K. Josephson, and K. Åström. A column-pivoting based strategy for monomial ordering in numerical gröbner basis calculations. In *The 10th European Conference on Computer Vision*, 2008. 3, 4

[4] M. Byröd, K. Josephson, and K. Åström. Fast and stable polynomial equation solving and its application to computer vision. *Int. Journal of Computer Vision*, 84(3):237–255, 2009. 1, 3

[5] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer, 2007. 3

[6] D. Eisenbud, D. R. Grayson, E. S. Michael, and B. Sturmfels, editors. *Computations in algebraic geometry with Macaulay 2*. Springer-Verlag, 2001. 6

[7] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *Computer Vision–ECCV 2012*, pages 738–751. Springer, 2012. 1

[8] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 383–390. IEEE, 2011. 1, 6, 7

[9] K. Josephson and M. Byröd. Pose estimation with radial distortion and unknown focal length. In *Proc. Conf. Computer Vision and Pattern Recognition, San Fransisco, USA*, 2009. 6

[10] Y. Kuang and K. Åström. Numerically stable optimization of polynomial solvers for minimal problems. In *Computer Vision–ECCV 2012*, pages 100–113. Springer, 2012. 1

[11] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *Proc. 10th European Conf. on Computer Vision, Marseille, France*, 2008. 1, 3, 6, 7, 8

[12] Z. Kukelova and T. Pajdla. A minimal solution to radial distortion autocalibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2410–2422, 2011. 6

[13] F. M. Mirzaei and S. I. Roumeliotis. Globally optimal pose estimation from line correspondences. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5581–5588. IEEE, 2011. 1, 5, 7, 8

[14] O. Naroditsky and K. Daniilidis. Optimizing polynomial solvers for minimal geometry problems. In *ICCV*, pages 975–982, 2011. 1

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

[16] C. Olsson, F. Kahl, and M. Oskarsson. Branch and bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009. 5, 7

[17] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, Apr. 2005. 1

[18] L. Zhang, C. Xu, K.-M. Lee, and R. Koch. Robust and efficient pose estimation from line correspondences. In *ACCV 2012*, pages 217–230. Springer, 2013. 7

[19] Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström, and M. Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *2013 IEEE International Conference on Computer Vision (ICCV)*. 2013. 1