

Optimizing Polynomial Solvers for Minimal Geometry Problems

Oleg Naroditsky and Kostas Daniilidis
University of Pennsylvania
Philadelphia, PA

{narodits, kostas}@cis.upenn.edu

Abstract

In recent years polynomial solvers based on algebraic geometry techniques, and specifically the action matrix method, have become popular for solving minimal problems in computer vision. In this paper we develop a new method for reducing the computational time and improving numerical stability of algorithms using this method. To achieve this, we propose and prove a set of algebraic conditions which allow us to reduce the size of the elimination template (polynomial coefficient matrix), which leads to faster LU or QR decomposition. Our technique is generic and has potential to improve performance of many solvers that use the action matrix method. We demonstrate the approach on specific examples, including an image stitching algorithm where computation time is halved and single precision arithmetic can be used.

1. Introduction

Since the introduction of the Gröbner basis methods to computer vision by Stewenius [17, 15], numerous problems in geometry have been expressed as polynomial systems and solved. Minimal problems are particularly important in structure from motion, absolute pose estimation and feature-based image registration (stitching) because they construct hypotheses from minimal data, which, in turn, minimizes the probability of including an outlier in a hypothesis of a RANSAC-based process [7, 14]. Before the proliferation of these solvers researchers often relied on linear algorithms which needed larger than minimal sets of points and hence considerably longer and less robust sampling process in RANSAC.

The first efficient solver for a minimal problem in computer vision, introduced by Nister [13] for the five-point relative pose, used a hand-crafted Gröbner basis solver. Since then Gröbner basis solvers were devised for a number of minimal problems, such as the solutions to autocalibration of radial distortion [10, 6], relative pose with unknown focal length [1], and infinitesimal camera motion [16]. Panoramic

image stitching with unknown focal length has been solved [2] as well and will serve as the main demonstration of our approach. Some non-minimal problems have also been solved, such as optimal three-view triangulation [17, 3].

Since its introduction, several improvements to the Gröbner basis method (also known in the community as the *action matrix* method) have been devised to address its numerical shortcomings [4] and ease of use [8]. While the action matrix method has been able to provide satisfactory (and in many cases the only) solutions to a range of problems, it still suffers from two main drawbacks: computational complexity and numerical accuracy.

Once an elimination template for a problem has been constructed, the main computational steps of the action matrix approach are: matrix decomposition (such as LU or QR) of a polynomial coefficient matrix in order to express one set of monomials in terms of another and construct the action matrix, and an eigenvalue decomposition of the action matrix to find the solutions. The size of the eigenvalue problem is typically related to the number of solutions, so the computational complexity of that part cannot be significantly reduced. The matrix used in the LU decomposition typically comes out of the structure of underlying problem and can be quite large. In this paper we show that under some conditions, it is possible to reduce significantly the size of this matrix while guaranteeing that the algebraic structure of the problem is not affected. Since matrix decompositions are typically $O(n^3)$ operations, a reduction in its size produces a significant performance gain.

The numerical stability issues with the action matrix method are also well known. Several methods have been proposed to address this problem directly, such as using a redundant solving basis and basis selection by SVD or QR decomposition [4], but the underlying problem is the size of the LU or QR decomposition, and matrix conditioning issues associated with it. We will consider a specific problem of 3D panorama stitching with unknown focal length and radial distortion to demonstrate experimentally that after using our method to reduce the size of the matrix decomposition, numerical accuracy improves significantly. We

show that the improved method is accurate even with single-precision arithmetic, which makes a fast implementation on a smartphone possible.

The effectiveness of our approach is tested on real imagery. We will show that in the case of RANSAC-based image stitching, our improved template can achieve good single-precision accuracy, while the state-of-the-art method fails to produce a solution due to round-off errors.

2. The Action Matrix Method

We will briefly outline the action matrix method to introduce the notation (see also [5, 9]). Let $F = \{f_1 \dots f_n\}$ be a set of polynomials in variables $\mathbf{x} = x_1 \dots x_l$. The polynomials F generate an ideal I , and we assume there exists a finite-dimensional quotient space $\mathbb{R}[\mathbf{x}]/I$, thus the system F has a finite number of zeros. The aim of the action matrix method is to construct a matrix in the quotient ring space that multiplies polynomials by x_k , i.e. matrix A such that $x_k \mathbf{v}^\top \mathbf{X} = A^\top \mathbf{v}^\top \mathbf{X}_B$, where $\mathbf{v}^\top \mathbf{X}_B \in \mathbb{R}[\mathbf{x}]/I$ and \mathbf{X}_B is a monomial basis for $\mathbb{R}[\mathbf{x}]/I$. The solutions to $F = 0$ are extracted from the eigenvalues of A . Since this expression is valid for any polynomial in the quotient space we obtain

$$x_k \mathbf{X}_B = A^\top \mathbf{X}_B, \quad (1)$$

which suggests that we can construct the action matrix by finding a set of polynomial equations which express the monomials $x_k \mathbf{X}_B$ in terms of the basis monomials.

We identify the *basis, required* and *extra* monomial subsets. The basis monomials \mathcal{B} constitute a linear basis for the quotient ring $\mathbb{R}[\mathbf{x}]/I$ (although \mathcal{B} can include other monomials when the *redundant solving basis* method is used, see [5], Section 4), and the required monomials are $\mathcal{R} = x_k \mathcal{B} \setminus \mathcal{B}$. The *extra* monomials \mathcal{E} are the remaining monomials.

The solver generation usually proceeds as follows: the initial polynomial system F is extended with additional equations from the ideal $I = \langle F \rangle$. These equations are found by multiplying the equations from F by monomials, starting with lowest order. The final system has the following form

$$C\mathbf{X} = 0, \quad (2)$$

where C is a matrix of polynomial coefficients of size $n \times m$ and \mathbf{X} is the vector of all monomials in the extended system.

We rearrange the columns of the system as follows

$$C\mathbf{X} = \begin{bmatrix} C_{\mathcal{E}} & C_{\mathcal{R}} & C_{\mathcal{B}} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{B}} \end{bmatrix} = 0. \quad (3)$$

If a sufficient number of polynomials was added to the original system F , the action matrix can be found by performing LU decomposition of C . The starting point for our

optimization is such a system with basis, required and extra monomials identified. We refer to this as an *elimination template*.

3. The Conditions for Template Simplification

First, we show under what circumstances we can separate the system $C\mathbf{X}$ into two sets of columns and stay in the ideal after elimination of one of the sets. This is a technical condition required for Lemma 2. We denote by $C_{\setminus j} \mathbf{X}_{\setminus j}$ the system $C\mathbf{X}$ with column \mathbf{c}_j and the corresponding monomial \mathbf{x}_j removed.

Lemma 1. *Let C be a full-rank matrix of polynomial coefficients and \mathbf{X} a vector of monomials, such that $C\mathbf{X}$ is a set of n polynomials which generate an ideal I . Let \mathbf{c}_j be a column of C . Let $C_{\setminus j} = LU$ be the LU decomposition of $C_{\setminus j}$. Then the polynomials*

$$U\mathbf{X}_{\setminus j} + L^{-1}\mathbf{c}_j\mathbf{x}_j \quad (4)$$

are also in I .

Proof. The matrix L^{-1} is a product of elementary lower triangular matrices (see [12] p.142), and thus the transformation L represents only row operations. Since applying row operations to C will keep the equations in the ideal, the following statements are true:

$$\begin{aligned} C\mathbf{X} &\in I \\ C_{\setminus j}\mathbf{X}_{\setminus j} + \mathbf{c}_j\mathbf{x}_j &\in I \\ LUC_{\setminus j}\mathbf{X}_{\setminus j} + \mathbf{c}_j\mathbf{x}_j &\in I \\ L^{-1}(LUC_{\setminus j}\mathbf{X}_{\setminus j} + \mathbf{c}_j\mathbf{x}_j) &\in I \\ U\mathbf{X}_{\setminus j} + L^{-1}\mathbf{c}_j\mathbf{x}_j &\in I, \end{aligned} \quad (5)$$

which is the needed result. \square

This lemma will be used to show that the equations stay in the ideal even after certain columns are removed. We now show that given C , it may be possible to compute the LU decomposition of a matrix smaller than C . We will now state and prove a condition under which we can remove a row and a column of C and still maintain the properties needed for successful elimination.

Lemma 2. *If there exists a column \mathbf{c}_j of the polynomial coefficient matrix C that satisfies the following two properties:*

1. *It corresponds to one of the extra monomials $\mathbf{X}_{\mathcal{E}}$, i.e. $\mathbf{c}_j \in C_{\mathcal{E}}$*
2. *It can be expressed as a linear combination of the first $m - |\mathcal{R}|$ columns*

then the matrix $C_{\setminus j}$ can be used for expressing the required monomials \mathcal{R} in terms of basis monomials \mathcal{B} via its LU decomposition.

Proof. The need for property 1 is clear since discarding a column corresponding to either a required or basis monomial will result in the corresponding terms missing from the equations.

To show the sufficiency of property 2, let us move \mathbf{c}_j to the right-hand side in the original system:

$$\begin{bmatrix} C_{\mathcal{E}\setminus j} & C_{\mathcal{R}} & C_{\mathcal{B}} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}\setminus j} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{B}} \end{bmatrix} = -\mathbf{c}_j \mathbf{x}_j. \quad (6)$$

After LU factorization of the left-hand side matrix $C_{\setminus j} = LU$ the system becomes

$$L \begin{bmatrix} U_{\mathcal{E}\setminus j} & C_{\mathcal{R}1} & C_{\mathcal{B}1} \\ 0 & U_{\mathcal{R}} & C_{\mathcal{B}2} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}\setminus j} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{B}} \end{bmatrix} = -\mathbf{c}_j \mathbf{x}_j, \quad (7)$$

where $U_{\mathcal{E}\setminus j}$ and $U_{\mathcal{R}}$ are upper-triangular matrices. Multiplying by L^{-1} from the left, we obtain

$$\begin{bmatrix} U_{\mathcal{E}\setminus j} & C_{\mathcal{R}1} & C_{\mathcal{B}1} \\ 0 & U_{\mathcal{R}} & C_{\mathcal{B}2} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}\setminus j} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{B}} \end{bmatrix} = -L^{-1} \mathbf{c}_j \mathbf{x}_j. \quad (8)$$

Where the polynomials formed by the rows belong to the original ideal by Lemma 1.

Since we need exactly $|\mathcal{R}|$ equations where the basis monomials are expressed in terms of required monomials, the last $|\mathcal{R}|$ elements of the vector $L^{-1} \mathbf{c}_j \mathbf{x}_j$ must be 0 (then the corresponding equations will not include any terms with \mathbf{x}_j). Let us show that this is implied by property 2.

Let us rewrite property 2 as

$$\mathbf{c}_j = a_1 \mathbf{c}_1 + \dots + a_{j-1} \mathbf{c}_{j-1} + a_{j+1} \mathbf{c}_{j+1} + \dots + a_{m-|\mathcal{R}|} \mathbf{c}_{m-|\mathcal{R}|}, \quad (9)$$

where \mathbf{c}_k are the first $m - |\mathcal{R}|$ columns of $C_{\setminus j}$. The lower triangular matrix L^{-1} acts on \mathbf{c}_k as follows: it annihilates elements below k (since $L^{-1} C_{\setminus j}$ is an upper triangular matrix). Thus the right hand side of the equation is

$$L^{-1} \mathbf{c}_j \mathbf{x}_j = a_1 \begin{bmatrix} \bullet \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mathbf{x}_j + \dots + a_{m-|\mathcal{R}|} \begin{bmatrix} \bullet \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mathbf{x}_j, \quad (10)$$

where the last vector has zeros in the last $|\mathcal{R}|$ positions. \square

The above lemma allows construction of a smaller matrix on the left-hand side of the system (2) by moving terms to the right-hand side and guarantees that the right-hand side will be zero in the last $|\mathcal{R}|$ equations.

While removing whole rows (equations) will keep the remainder in I , the ideal generated by the resulting set will be smaller if the lowest order generators are removed. We can now define excess columns as well as excess row-column pairs, removal of which will not affect the elimination.

Definition 1. An excess column \mathbf{c}_j of C is a column that satisfies Lemma 2, and an excess pair of C is a row \mathbf{r}_i of C and a column \mathbf{c}_j , such that \mathbf{c}_j is a column of C with \mathbf{r}_i removed that satisfies Lemma 2.

The proposed optimization of elimination templates will make use of the Lemma 2 to find excess columns and pairs. At this point it is clear how we should proceed: we will first test each column to see if it can be safely moved to the right-hand side of the equation $C\mathbf{X} = 0$ and then, when no excess columns are found, we test all row-column combinations. The progressively smaller matrix will retain the full left and right hand sides, such that at any point all the equations in it are in the ideal. Since we will be operating with specific instances of the problems (symbolic elimination is impossible for most templates), we will have to verify the resulting template thoroughly for numerical stability and repeat the process as necessary to achieve a good template.

We now proceed to illustrate this process step-by-step with an example.

4. Example: Three-point Panorama Stitching

As an example, we demonstrate how to reduce the size of the elimination template for the problem of estimating panoramic stitching parameters in the case of unknown focal length and radial distortion [2]. In this problem we estimate the rotation between two views and the common focal length and radial distortion coefficient. We only present the problem formulation and refer the reader to that paper for solution details. Our optimization is implemented as a modification of an existing implementation available from Lund University.¹

4.1. Polynomial Model

The two cameras have square pixels, zero skew and the principal point in the center of the image, thus the calibration matrix $K = \text{diag}(f, f, 1)$, where f is the focal length. The two camera views also share a common origin in the stitching scenario and thus $P_1 = K[I \ 0]$ and $P_2 = K[R \ 0]$, where $R \in \text{SO}(3)$. We now arrive at the following relation between world point \mathbf{U} and image point \mathbf{u} :

$$z_1 \mathbf{u} = K\mathbf{U}, \quad z_2 \mathbf{u} = KR\mathbf{U}, \quad (11)$$

¹The code is found at <http://www.maths.lth.se/vision/downloads/data/stitching3pt.zip>. Our optimized code can be found at http://www.cis.upenn.edu/~narodits/Site_3/Solver_Optimization.html.

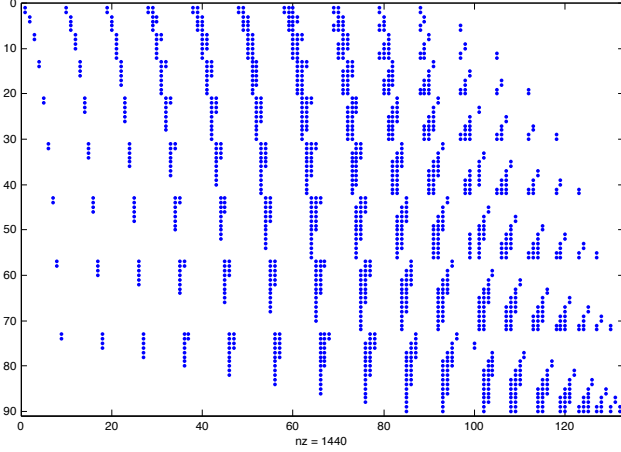


Figure 1. Structure of the template as described in [2].

where z_1 and z_2 are the depths. The dependence on the depths can be removed by rewriting the constraints as

$$\frac{\langle K^{-1}\mathbf{u}_{1j}, K^{-1}\mathbf{u}_{1k} \rangle^2}{|K^{-1}\mathbf{u}_{1j}|^2 |K^{-1}\mathbf{u}_{1k}|^2} = \frac{\langle \mathbf{U}_j, \mathbf{U}_k \rangle}{|\mathbf{U}_j|^2 |\mathbf{U}_k|^2} = \frac{\langle K^{-1}\mathbf{u}_{2j}, K^{-1}\mathbf{u}_{2k} \rangle^2}{|K^{-1}\mathbf{y}_{2j}|^2 |K^{-1}\mathbf{u}_{2k}|^2} \quad (12)$$

These constrains are augmented with radial distortion model $|v| = (1 + \lambda|v|^2)|u|$. The normalized image point can now be expressed as

$$\mathbf{u} \sim \mathbf{v} + \lambda[0 \ 0 \ v_1^2 + v_2^2]^\top \quad (13)$$

By substituting (13) into (12), squaring to remove the square roots and multiplying through by the denominators, we obtain a polynomial in degree 3 in f^2 , and degree 6 in λ . Using constraints from three point correspondences yields a system with 2 equations and 18 solutions [2].

4.2. Eliminating Excess Columns

The template for this problem consists of 90 equations in 132 monomials, and is solved using the redundant solving basis method and QR decomposition of a 90×100 matrix in order to create a numerically stable basis. The structure of the template is shown in Figure 1. It is arranged such that $C = [C_{\mathcal{E}} \ C_{\mathcal{R}} \ C_{\mathcal{B}}]$, where $|\mathcal{E}| = 100$, $|\mathcal{R}| = 7$ and $|\mathcal{B}| = 25$. Thus the system has 25 solutions and will be solved via eigenvalue decomposition of a 25×25 action matrix.

We will now show how to use the results in Section 3 to reduce the size of this template. One way to identify excess columns present in the initial template is via Gaussian elimination with partial pivoting. The eliminated matrix has the structure shown in Figure 2. It is important to note at this point that since symbolic elimination of such a large matrix is not practical, the matrix structure above is from a

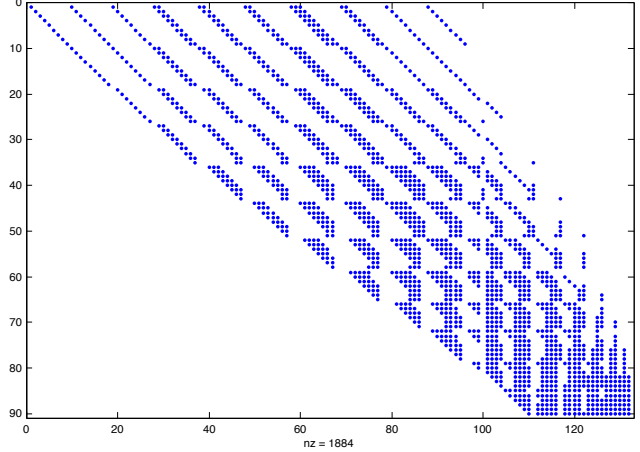


Figure 2. Structure of elimination template after Gaussian elimination.

“typical” numeric example. While cancellation pattern and structure of a symbolic Gaussian elimination will always be the same (since the relationships between all matrix entries are fixed), a numeric elimination may encounter problems with precision (especially in the lower right part of the matrix), and thus give different structure for different instances of the problem. We must try many instances until a stable row-echelon form of C is found.

In the structure above we can immediately observe columns which have zero pivots. These columns, namely [19, 28, 38, 39, 48, 49, 58, 59, 60, 69, 70, 79, 80, 88], correspond to excess columns and can each be expressed as a linear combination of the previous, extra columns, and thus by Lemma 2, they can be eliminated from the matrix. At this point the template matrix is 118×90 .

4.3. Eliminating Excess Pairs

For the second round of optimization, we will find excess pairs by testing each row and column. This is accomplished by systematically removing a row of C , and for each column corresponding to an \mathcal{E} monomial, computing the vector $L^{-1}\mathbf{c}_j x_j$, as in equation (8) and checking if the \mathcal{R} monomials have zero coefficients. If the number of zeros is greater than or equal to $|\mathcal{R}|$, then the row is removed and the column is moved to the right-hand side. We repeat this process on the resulting left and right hand side matrices until a smallest size left-hand side matrix is found. Once again, the numerical stability of this process is an issue, and a sufficient number of instances must be tried in order to ascertain the quality of the resulting template. This optimization round allowed us to create a stable template of size 54×77 with QR decomposition being performed on a 54×45 matrix. The structure of the optimized template is shown in Figure 3. In Appendix A, we list the complete set of excess rows and columns which can be used to reproduce our

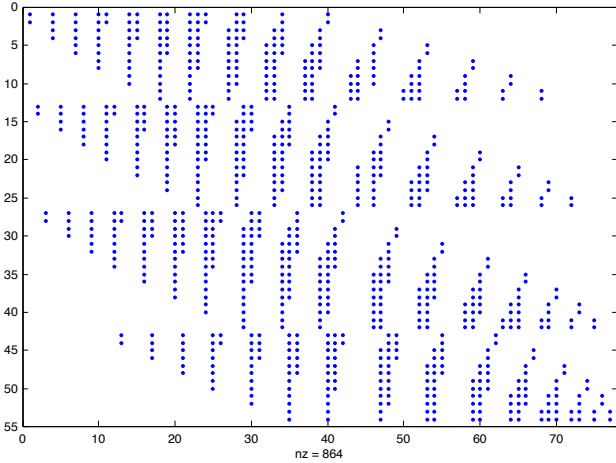


Figure 3. Structure of the template after optimization.

results.

4.4. Numerical Stability

The numerical stability of the resulting template is verified in Figures 4 and 5. We observe that the performance is almost identical in both the noise-free and noisy cases on a large set of randomly generated configurations. In terms of computational performance, the average QR decomposition time decreased from 1.1ms to 0.24ms, which translates into a doubling of overall performance when matrix inversion and eigenvalue decomposition are taken into account.

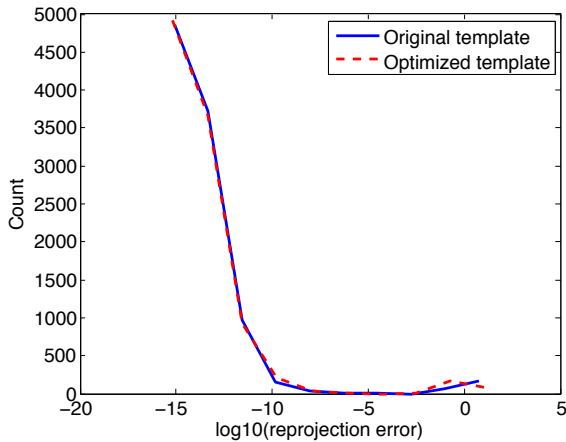


Figure 4. Noise free, double precision experiment for the three-point stitch problem. Comparison of the orders of magnitude of reprojection errors between solutions to 10^5 noise-free instances. Reduced template is dashed red and the original template is solid blue.

Aside from speed, the other significant advantage of our new template is its numerical stability under single precision arithmetic. Due to the smaller size of our template, the round-off errors are not as significant. The comparison

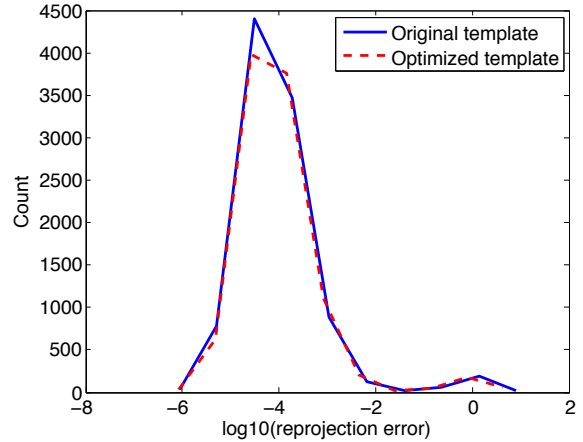


Figure 5. Noisy, double precision experiment for the three-point stitch problem. Comparison of the orders of magnitude of reprojection errors between solutions to 10^5 instances. The points are contaminated with 0.01 standard deviation noise in the normalized image plane. The dashed, red plot was generated with the reduced template and solid, blue with the original. The solutions obtained by both templates are very close, which shows that we did not compromise numerical stability of the algorithm by reducing the size of the template.

of the two templates in single precision is shown in Figure 6. The original template is not usable in single-precision arithmetic with a 4300 of 10000 cases failing to produce the correct solution. In contrast, the reduced template failed in only 224 cases. The single precision implementation used single precision for QR decomposition, matrix division and eigenvalue decomposition. It is clear that for the original template, it is the 90×100 QR decomposition that turned out to be unstable.

4.5. Experimental Results

Finally, we verify performance of the reduced template on real images. In Figures 7 and 8 we show the panoramas generated with our new template with single precision arithmetic. The procedure to generate these results was as follows: 1) detect SIFT [11] features on the two images and match them, 2) run a RANSAC process on the subsets of the matches and output the best hypothesis, and 3) warp the images using the hypothesis.

On the feature points shown, both double precision implementations produced similar results, however, the single precision implementations differed dramatically. Of the 300 generated hypotheses, the original template failed to find any solutions in 297 cases (the remaining 3 cases gave wrong solutions) and the reduced template only had 43 such failures (for the images in Figure 8).

Overall, we can conclude that the reduced template is stable and can be applied to real-world images, even on sin-

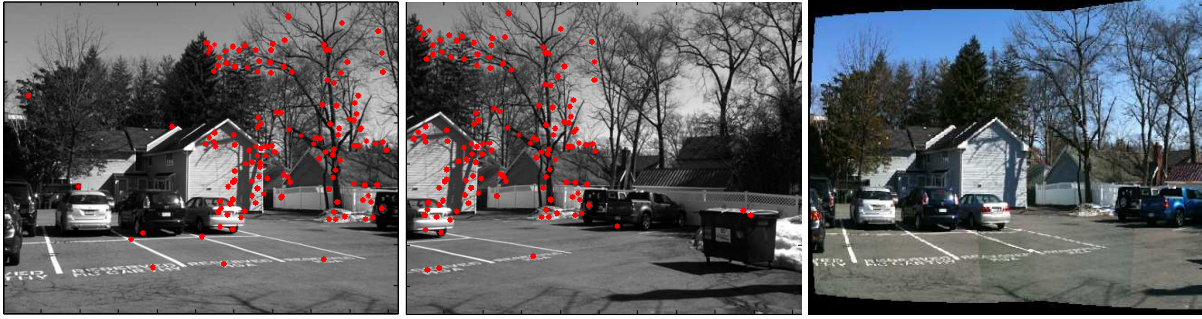


Figure 7. The matched SIFT features on the cell phone images and the resulting panorama generated with single precision arithmetic using the reduced template and RANSAC. The performance was similar for the double precision in both templates, however, the single precision original template did not produce a valid hypothesis after 200 iterations. No bundle adjustment or image blending was applied, hence the panorama is generated with a single three-point hypothesis.



Figure 8. The matched SIFT features on high quality images and the resulting panorama generated with single precision arithmetic using the reduced template and RANSAC.

gle precision hardware, such as a smartphone.

5. Example: Optimal Three-view Triangulation

The problem of L2-optimal, three-view triangulation is a classic problem that is solved by the action matrix method. First solved by Stewenius *et. al* in [17] in 256-bit arithmetic, this problem was revisited by Byrod *et. al* [3] and solved with double precision arithmetic. With its 47-dimensional linear basis, this problem is difficult to optimize, however, we still manage to improve performance slightly. We base our implementation on the freely available version from Byrod.²

5.1. Polynomial Model

The optimal three-view triangulation problem is not a minimal problem (minimal number of views is 2), but it is a classic problem solved by the action matrix method. We briefly review the model. Given three camera matrices

$\mathbf{P}_i = [R_i \ t_i]$, and three corresponding image points \mathbf{u}_i , the 3D point $\mathbf{U} = [u_1, u_2, u_3]^T$ must be computed such that it minimizing the objective function defined as the sum of squared reprojection errors:

$$F(\mathbf{U}) = \sum_{i=1}^3 d(\mathbf{P}_i \mathbf{U}, \mathbf{u}_i)^2. \quad (14)$$

The problem is simplified by setting all image points \mathbf{u}_i to be at the origin, and adjusting the camera matrices to compensate. This is done by rotating them around the camera's center of projection to get $\mathbf{P}'_i = [R_{\mathbf{u}_i} R_i \ t]$, where $R_{\mathbf{u}_i} \in SO_3$ is the rotation between \mathbf{u}_i and $[0, 0, 1]^T$ in the camera coordinate system. The objective is now $F(\mathbf{U}) = \sum_{i=1}^3 F_i(\mathbf{U})$, where reprojection errors $F_i(\mathbf{U})$ are given in terms of the rows of the camera matrices $\mathbf{P}'_i = [P_i^1 \ P_i^2 \ P_i^3]^T$:

$$F_i(\mathbf{U}) = \frac{(P_i^1 \mathbf{U})^2 + (P_i^2 \mathbf{U})^2}{(P_i^3 \mathbf{U})^2} \quad (15)$$

which is the squared length of the image vector after re-projection.

Optimality requires that $\frac{dF}{du_i} = 0$ for each component u_i of \mathbf{U} . After differentiating with respect to u_i . When $i \neq j$,

²The code is at <http://www.maths.lth.se/vision/downloads/data/optimal.tvt.zip>

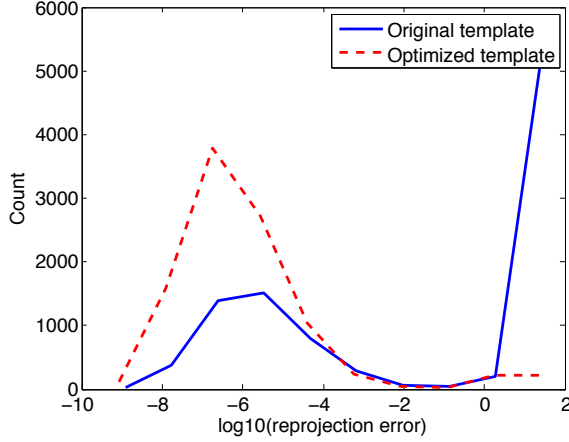


Figure 6. A noise-free, single precision experiment for the three-point stitch problem. The graph shows the comparison of the orders of magnitude of reprojection errors between solutions to 10^5 instances of the problem. The dashed, red plot was generated with the reduced template and solid, blue with the original. This plot demonstrates that only our reduced template can be used with single precision since in about 4300 cases the original template fails completely.

the derivative is [15]

$$\frac{dF_i}{du_j} = 2 \frac{P_i^1(j)P_i^1\mathbf{U} + P_i^2(j)P_i^2\mathbf{U}}{u_j^2}, \quad (16)$$

where $P_i^k(j)$ is the j th element of P_i^k . When $i = j$, we have

$$\frac{dF_j}{du_j} = \frac{2(P_j^1(j)P_j^1\mathbf{U} + P_j^2(j)P_j^2\mathbf{U})u_j - (P_j^1\mathbf{U})^2 + (P_j^2\mathbf{U})^2}{u_j^3} \quad (17)$$

When these derivatives are written out in terms of u_j for $j = 1 \dots 3$ and brought under the same denominator to form polynomial equations, we end up with 3 equations of degree 6.

5.2. Template Optimization

After the addition of a sufficient number of equations, the elimination template for this problem has size 225×209 . We reduced the size to 154×204 . We begin by rearranging the columns of the coefficient matrix such that $C = [C_E \ C_R \ C_B]$. We then remove rows, one-by-one and test the template for stability, as suggested in [8].

This allows us to remove rows

[1 . . . 43, 49, 122, 124, 125, 127, 128, 130,
133, 167, 169, 170, 171, 172, 199, 200, 202, 204,
206, 209, 210, 213, 214, 218, 222].

We now apply our method to remove five additional excess pairs from the resulting matrix. We remove rows [21 , 79 , 80 , 81, 117], and columns [1, 2, 3, 55 , 56]. The number of pairs removed is only 5, however, this problem has 50 basis monomials and 31 required monomials, so the task of choosing Lemma 2 compliant columns is difficult. It is still possible that a smaller template lurks inside this one, but finding it might be a difficult combinatorial problem.

We use simulated data to show once again that this removal has little effect on the results in double precision arithmetic. We plot the results in Figures 9 and 10. Single precision appears to be far out of reach for this problem.

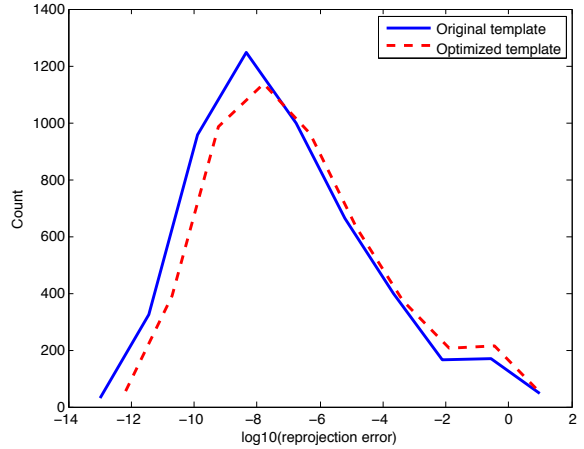


Figure 9. Optimal three-view triangulation experiment with no noise. We compare the orders of magnitude of error in triangulated 3D point $\log_{10}(\|X - \hat{X}\|_2)$. We used the standard basis (vvt_solve_std.m script), and not the QR method. Note that due to the nature of the problem, the error is much higher than in the case of 3D panorama stitching.

6. Example: The 3+1 Problem

We have also applied our method to the problem of estimating relative pose in the case of two known orientation angle, which we call the "3+1 problem." In our technical report we present an action matrix based solution to the problem and optimize it using this method. In that case, the template size goes from 36×46 to 21×25 . The exact models, the numeric performance (including the single precision case) and applications are discussed in the report.³

³The report and code are at http://www.cis.upenn.edu/~narodits/Site_3/3+1.html.

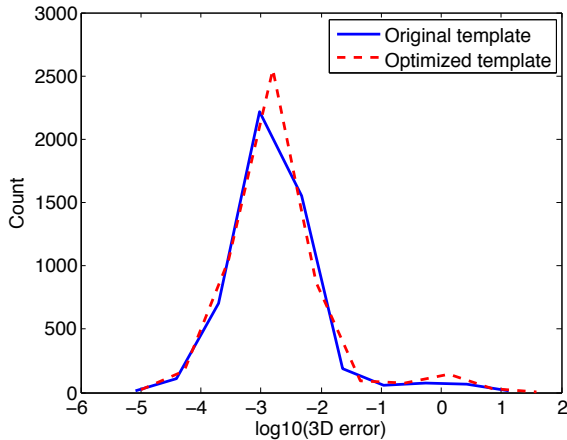


Figure 10. Optimal three-view triangulation experiment with noise. We compare the orders of magnitude of error in triangulated 3D point $\log_{10}(\|X - \hat{X}\|_2)$ between the two templates. The standard deviation of the noise in the points in camera coordinates was 0.01. The results are for the standard basis (tv_solve_std.m script), and not the QR method. The results are again very similar despite being computed with a smaller template.

7. Conclusions and Future Work

In this paper we developed a new method for reducing the size of the action matrix template and proved some properties of the columns of the template matrix. We showed that under some conditions, rows and columns of the matrix can be removed a priori, resulting in improved speed.

We produced a real example of algorithms which benefits substantially from our approach both in terms of speed and numerical stability, namely the three-point panoramic stitching and the “3+1” algorithm. We believe that this generic approach can be used on other algorithms as well, making the action matrix method an even more attractive way to solve geometry problems in computer vision.

We will investigate if this method can be improved by applying Lemmas 1 and 2 at the template generation stage with the coefficients in $\mathbb{Z}/p\mathbb{Z}$, yielding an exact solution.

8. Acknowledgement

The authors gratefull for support through the following grants: NSF-IIP-0742304, NSF-IIP- 0835714, NSF-OIA-1028009, ARL MAST-CTA W911NF-08-2-0004, and ARL RCTA W911NF-10-2-0016, NSF-DGE-0966142.

A. Three-point Panoramic Stitching Excess Rows and Columns

In this appendix we list the rows and columns removed from the original 90×132 matrix constructed in the function setup_3pt.m of the aforementioned code. After arranging the matrix as $C = [C_E \ C_R \ C_B]$, we remove the columns $[1 \dots 5, 9 \dots 14, 18 \dots 23, 27 \dots 32, 37 \dots 42,$

$47 \dots 52, 57 \dots 62, 68 \dots 72, 78 \dots 80, 88]$ and rows $[1 \dots 30, 73 \dots 78]$. After this, the elimination proceeds as before, except we must adjust the indices `m_ind` to compensate for a smaller matrix.

References

- [1] M. Bujnak, Z. Kukelova, and T. Pajdla. A general solution to the p4p problem for camera with unknown focal length. *Proc. of Conf. Computer Vision and Pattern Recognition*, 2008.
- [2] M. Byröd, M. Brown, and K. Åström. Minimal solutions for panoramic stitching with radial distortion. *Proc. of British Machine Vision Conference, London, United Kingdom*, 2009.
- [3] M. Byrod, K. Josephson, and K. Astrom. Fast optimal three view triangulation. *In Proc. of Asian Conference on Computer Vision*, Jan 2007.
- [4] M. Byrod, K. Josephson, and K. Astrom. Improving numerical accuracy of grobner basis polynomial equation solver. *In Proc. of International Conference on Computer Vision*, 2007.
- [5] M. Byrod, K. Josephson, and K. Åström. Fast and stable polynomial equation solving and its application to computer vision. *International Journal of Computer Vision*, Jan 2009.
- [6] M. Byrod, Z. Kukelova, K. Josephson, and T. Pajdla. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. *In Proc. of Conf. on Computer Vision and Pattern Recognition*, Jan 2008.
- [7] M. Fischler and R. Bolles. Random sample consensus. *Communications of the ACM*, Jan 1981.
- [8] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. *In Proc. of European Conference on Computer Vision*, Jan 2008.
- [9] Z. Kukelova, M. Byröd, K. Josephson, and T. Pajdla. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. *Computer Vision and Image Understanding*, Jan 2008.
- [10] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. *Proc. of Conf. Computer Vision and Pattern Recognition*, 2007.
- [11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Jan 2004.
- [12] C. Meyer. Matrix analysis and applied linear algebra. *SIAM*, Jan 2000.
- [13] D. Nister. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence*, Jan 2004.
- [14] D. Nister. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.
- [15] H. Stewenius. Gröbner basis methods for minimal problems in computer vision. *maths.lth.se*, Jan 2005.
- [16] H. Stewenius, C. Engels, and D. Nister. An efficient minimal solution for infinitesimal camera motion. *In Proc. of Conf. on Computer Vision and Pattern Recognition*, Jan 2007.
- [17] H. Stewenius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really? *In Proc. of International Conference on Computer Vision*, Jan 2005.