Fast Optimal Three View Triangulation

Byröd, Martin; Josephson, Klas; Åström, Karl

2008

Fast Optimal Three View Triangulation

Martin Byröd, Klas Josephson and Kalle Åström
Center for Mathematical Sciences,
Lund University, Lund, Sweden
{byrod, klasj, kalle}@maths.lth.se.

Abstract

We consider the problem of L_2-optimal triangulation from three separate views. Triangulation is an important part of numerous computer vision systems. Under gaussian noise, minimizing the L_2 norm of the reprojection error gives a statistically optimal estimate. This has been solved for two views. However, for three or more views, it is not clear how this should be done. A previously proposed, but computationally impractical, method draws on Gröbner basis techniques to solve for the complete set of stationary points of the cost function. We show how this method can be modified to become significantly more stable and hence given a fast implementation in standard IEEE double precision. We evaluate the precision and speed of the new method on both synthetic and real data. The algorithm has been implemented in a freely available software package which can be downloaded from the Internet.

1 Introduction

Triangulation, referring to reconstructing the 3D location of a point given its images in two or more known views, is a fundamental part of numerous computer vision systems. Albeit conceptually simple, this problem is not completely solved in the general case of n views and noisy measurements.

There exist fast and relatively robust methods based on linear least squares [9]. These methods are however sub-optimal and can in unfortunate situations yield very poor accuracy.

The best approach is instead to minimize the L_2 norm of the reprojection error, i.e. the sum of squares of the reprojection errors, which has been given a closed form solution in the case of two views [8].

In this paper, we propose to solve the problem of L_2 optimal triangulation from three views using a method introduced in [10], where the optimum was found by explicit computation of the complete set of stationary points of the likelihood function. This approach is similar to that of [8]. However, whereas the stationary points in the two view case can be found by solving a sixth degree polynomial in one variable, the three view case involves solving a system of three sixth degree equations in three unknowns with 47 solutions. Thus, we have to resort to more sofisticated techniques to tackle this problem.

Stewenius *et al.* used algebraic geometry and Gröbner basis techniques to analyse and solve the equation system. However, Gröbner basis calculations are known to be numerically challenging and they were forced to use emulated 128 bit precision arithmetics to get a stable implementation, which rendered their solution too slow to be of any practical value.

In this paper we develop the Gröbner basis approach further to improve the numerical stability and are able to give the Gröbner basis method a fast implementation using standard IEEE double precision.

Our main contributions are:

- A modified version of the Gröbner basis method for solving polynomial equation systems, here referred to as *the relaxed ideal method*, which trades some speed for a significant increase in numerical stability.
- An effecient C++ language implementation of this method applied to the problem of three view triangulation.

The source code for the methods described in this paper is freely available for download from the Internet[2].

2 Three View Triangulation

We assume a linear pin-hole camera model, i.e. projection in homogeneous coordinates is done according to $\lambda_i x_i = P_i X$, where P_i is the camera matrix for view i, x_i is the image coordinates, λ_i

is the depth and X is the 3D coordinates of the world point to be determined.

We minimise the L_2 norm of the reprojection errors. Since we are free to choose coordinate system in the images, we place the three image points at the origin in their respective image coordinate systems. We thus obtain the following cost function over X:

$$\varphi(X) = \Sigma_{k=1}^{3} \frac{(P_{k1}X)^2 + (P_{k2}X)^2}{(P_{k3}X)^2}, \qquad (1)$$

where e.g. P_{k3} refers to row 3 of camera k. We calculate the complete set of stationary points of $\varphi(X)$ by solving $\nabla\varphi(X) = 0$. The explicit derivatives can easily be calculated, but we refrain from writing them out here. Differentiating and multiplying through with the denominators produces three sixth degree polynomial equations in the three unknowns $X = [\ X_1 \quad X_2 \quad X_3\ ]$. The remainder of the theoretical part of the paper is devoted to the problem of solving these equations.

3 Gröbner Basis Techniques

We now give an outline of how Gröbner basis techniques can be used for solving systems of multivariate polynomial equations. The general theory of multivariate polynomials over any field is algebraic geometry. See e.g. [6] for a good introduction.

The overall goal is to find the set of solutions to a system $f_1(\mathbf{x}) = 0, \ldots, f_m(\mathbf{x}) = 0$ of m polynomial equations in n variables. The polynomials $f_1, \ldots, f_m$ generate an *ideal* I in $\mathbb{C}[\mathbf{x}]$, the ring of multivariate polynomials in $\mathbf{x} = (x_1, \ldots, x_n)$ over the field of complex numbers.

To find the roots of this system we study the quotient ring $\mathbb{C}[\mathbf{x}]/I$ of polynomials modulo I. If the system of equations has r roots, then $\mathbb{C}[\mathbf{x}]/I$ is a linear vector space of dimension r. In this ring, multiplication with x_k is a linear mapping. The matrix $\mathbf{m}_{x_k}$ representing this mapping (in some basis) is referred to as the action matrix. From algebraic geometry it is known that the zeros of the equation system can be obtained from the action matrix by using the fact that the vector of monomials spanning $\mathbb{C}[\mathbf{x}]/I$ evaluated at a zero of I is an eigenvector of $\mathbf{m}_{x_k}^t$.

$\mathbb{C}[\mathbf{x}]/I$ is a set of equivalence classes and to perform calculations in this space we need to pick representatives for the equivalence classes. A Gröbner basis G for I is a special set of generators for I with the property that it lets us compute a well defined representative for each equivalence class. We now turn to the numerical computation of a Gröbner basis.

4 Numerical Computations

There is a general method for constructing a Gröbner basis known as *Buchberger's algorithm* [5]. The idea is to succesively eliminate leading monomials from the equations by selecting polynomials pairwise and multiplying them by suitable monomials to be able to eliminate the least common multiple of their respective leading monomials. The algorithm stops when any new element from I reduces to zero upon multivariate polynomial division with the elements of G.

Buchberger's algorithm works perfectly under exact arithmetic. However, in floating point arithmetic it becomes extremely difficult to use due to accumulating round off errors. In Buchberger's algorithm, adding equations and eliminating is completely interleaved. We aim for a process where we first add all equations we will need and then do the full elimination in one go, in the spirit of the f_4 algorithm [7]. This allows us to use methods from numerical linear algebra such as pivoting strategies and QR factorization to circumvent (some of) the numerical difficulties.

This approach is made possible by first studying a particular problem using exact arithmetic to determine the number of solutions and what total degree we need to go to. Using this information, we hand craft a set of monomials which we multiply our original equations with to generate new equations. We stack the coefficients of our expanded set of equations in a matrix $\mathbf{C}$ and write our equations as

$$\mathbf{C}\boldsymbol{\varphi} = 0, \qquad (2)$$

where $\boldsymbol{\varphi}$ is a vector of monomials. Putting $\mathbf{C}$ on reduced row echelon form then gives us the reduced minimal Gröbner basiss. In the next section we go in to the details of constructing a Gröbner basis for the three view triangulation problem.

4.1 Gröbner Basis Construction

From Section 2, we obtained three sixth degree polynomial equations in $X = [X_1, X_2, X_3]$. To get a Gröbner basis as outlined in Section 4, after some preliminary steps, we multiply with monomials creating 225 equations in 209 different monomials of total degree up to nine. We then put the 225 by 209 matrix $\mathbf{C}$ on reduced row echelon form. See [4, 10] for details on the construction of the coefficient matrix.

The row echelon part turns out to be a delicate task due to generally very poor conditioning. In fact, the conditioning is often so poor that roundoff errors in the order of magnitude of machine epsilon

(approximately 10^{-16} for doubles) yield errors as large as 10^2 or more in the final result. This is the reason one had to resort to emulated 128 bit numerics in [10].

4.2 The Relaxed Ideal Method

After the equation adding steps described above, we have a set of equations which "tightly" describe the set of solutions and nothing more. By relaxing the constraints somewhat, possibly allowing some extra spurious solutions to enter the equations, we show that it is possible to get a significantly better conditioned problem. We do this by selecting a subset of the 225 equations. This choice is not unique, but a natural subset to use is the 55 equations with all possible 9th degree monomials as leading terms, since this is the smallest set of equations which directly gives us a Gröbner basis. We do this by QR factorization of the submatrix of $\mathbf{C}$ consisting of the 55 first columns followed by multiplying the remaining columns with Q^t. After these steps we pick out the 55 first rows of the resulting matrix. These rows correspond to 55 equations forming the relaxed ideal $I_{\text{rel}} \subset I$ which is a subset of the original ideal I. The set of eigenvalues computed from the action matrices for $\mathbb{C}[X]/I$ and $\mathbb{C}[X]/I_{\text{rel}}$ respectively are shown if Fig. 1. See [4] for an explanation and justification of this technique.

As shown in the experiments section, this improves the conditioning of the elimination step involved in the Gröbner basis computation considerably. The price we have to pay for this is performing an eigenvalue decomposition on a larger action matrix.

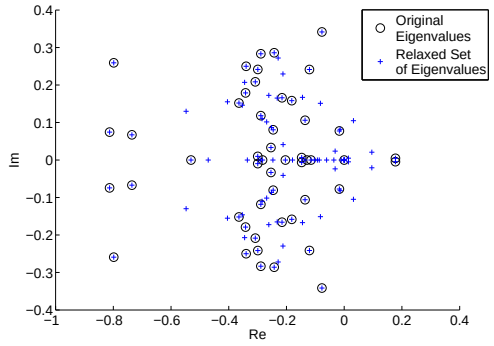

Figure 1: Eigenvalues of the action matrix using the standard method and the relaxed ideal method respectively, plotted in the complex number plane. The latter are a strict superset of the former.

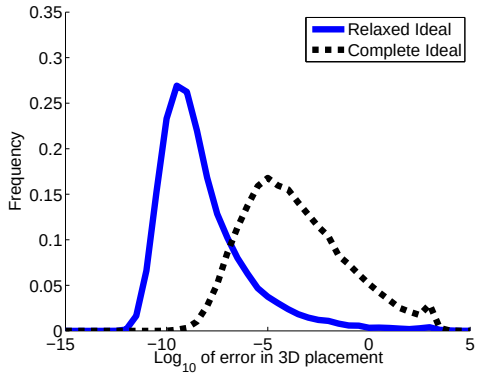

Figure 2: Histogram over the error in 3D location of the estimated point X. As is evident from the graph, extracting solutions from the smaller ideal yields a final result with considerably smaller errors.

5 Experimental Validation

The algorithm described in this paper has been implemented in C++ making use of LAPACK and BLAS [1] and the code is available for download from [2]. We have run the algorithm on both real and synthetically generated data using a 2.0 Ghz AMD Athlon X2 64 bit machine. With this setup, triangulation of one point takes approximately 60 milliseconds. This is to be contrasted with the previous implementation by Stewenius *et al.* [10], which needs 30 seconds per triangulation with their setup.

5.1 Synthetic Data

To evaluate the intrinsic numerical stability of our solver the algorithm has been run on 50.000 randomly generated test cases. World points were drawn uniformly from the cube $[-500, 500]^3$ and cameras where placed randomly at a distance of around 1000 from the origin with focallength of around 1000 and pointing inwards. We compare our approach to that of [10] implemented in double precision here referred to as the standard method. A histogram over the resulting errors in estimated 3D location is shown in Fig. 2. As can be seen, computing solutions of the smaller ideal yields an end result with vastly improved numerical precision. The error is typically around a factor 10^5 smaller with the new method.

5.2 A Real Example

Finally, we evaluate the algorithm under real world conditions. The Oxford dinosaur [3] is an image

sequence of a toy dinosaur shot on a turn table. The image sequence consists of 36 images and 4983 point tracks. For each point visible in three or more views we select the first, middle and last view and triangulate using these. This yields a total of 2683 point triplets to triangulate from. The image sequence contains some erroneus tracks which we deal with by removing any points reprojected with an error greater than two pixels in any frame. The whole sequence was processed in approximately 2.5 minutes and the resulting point cloud is shown in Fig. 3.

We have also run the same sequence using the previous method implemented in double precision, but the errors were too large to yield usable results. Note that [10] contains a successful triangulation of the dinosaur sequence, but this was done using extremely slow emulated 128 bit arithmetic yielding an estimated running time of 20h for the whole sequence.

Figure 3: The Oxford dinosaur reconstructed from 2683 point triplets using the method described in this paper. The reconstruction was completed in approximately 2.5 minutes.

6 Conclusions

In this paper we have shown how triangulation can be solved for the globally optimal L_2 estimate using Gröbner basis techniques. With the introduced method of the relaxed ideal, we have taken this approach to a state where it can now have practical value in actual applications.

Moreover, by this example we show that global optimisation by calculation of the stationary points using Gröbner basis techniques is indeed a possible way forward. This is particularly interesting since a large number of computer vision problems ultimately depend on some form of optimisation.

References

[1] Lapack - linear algebra package. http://www.netlib.org/lapack.

[2] Three view triangulation. http://www.maths.lth.se/∼byrod/downloads.html.

[3] Visual geometry group, university of oxford. http://www.robots.ox.ac.uk/∼vgg.

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

[5] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer Verlag, 1998.

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

[7] J.-C. Faugère. A new efficient algorithm for computing gröbner bases (f_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.

[8] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68:146–157, 1997.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] H. Stewénius, F. Schaffalitzky, and D. Nistér. How hard is three-view triangulation really? In *Proc. Int. Conf. on Computer Vision*, pages 686–693, Beijing, China, 2005.