Fast Optimal Three View Triangulation

Byröd, Martin; Josephson, Klas; Åström, Karl

Published in:
Lecture Notes in Computer Science

DOI:
10.1007/978-3-540-76390-1_54

2007

Fast Optimal Three View Triangulation

Martin Byröd, Klas Josephson and Kalle Åström
{byrod, klasj, kalle}@maths.lth.se

Center for Mathematical Sciences,
Lund University, Lund, Sweden

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 the act of 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 [1]. These methods are however sub-optimal. Moreover the linear least squares formulation does not have a clear geometrical meaning, which means that in unfortunate situations, this approch can yield very poor accuracy.

The most desirable, but non-linear, approach is instead to minimize the L_2 norm of the reprojection error, i.e. the sum of squares of the reprojection errors. The reason for this is that the L_2 optimum yields the maximum likelihood estimate for the 3D point under the assumption of independent gaussian noise on the image measurements [2]. This problem has been given a closed form solution[1] by Hartley and Sturm in the case of two views [2]. However, the approach of Hartley and Sturm is not straightforward to generalize to more than two views.

[1] The solution is actually not *entirely* on closed form, since it involves the solution of a sixth degree polynomial, which cannot in general be solved on closed form. Therefore one has to go by e.g. the eigenvalues of the companion matrix, which implies an iterative process.

In the case of n views, the standard method when high accuracy is needed is to use a two-phase strategy where an iterative scheme for non-linear least squares such as Levenberg-Marquardt (Bundle Adjustment) is initialised with a linear method [3]. This procedure is reasonably fast and in general yields excellent results. One potential drawback, however, is that the method is inherently local, i.e. it finds local minima with no guarantee of beeing close to the global optimum.

An interesting alternative is to replace the L_2 norm with the L_∞, norm cf. [4]. This way it is possible to obtain a provably optimal solution with a geomtrically sound cost function in a relatively efficient way. The drawback is that the L_∞ norm is suboptimal under gaussian noise and it is less robust to noise and outliers than the L_2 norm.

The most practical existing method for L_2 optimization with an optimality guarantee is to use a branch and bound approach as introduced in [5], which, however, is a computationally expensive strategy.

In this paper, we propose to solve the problem of L_2 optimal triangulation from three views using a method introduced by Stewenius *et al.* in [6], 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 Hartley and Sturm [2]. However, whereas the stationary points in the two view case can be found by solving a sixth degree polynomial in one variable, the easiest known formulation of 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. We show how computing the zeros of a relaxed ideal, i.e. a smaller ideal (implying a possibly larger solution set/variety) can be used to solve the original problem to a greater accuracy. Using this technique, we are able to give the Gröbner basis method a fast implementation using standard IEEE double precision. By this we also show that global optimization by calculation of stationary points is indeed a feasible approach and that Gröbner bases provide a powerful tool in this pursuit.

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[7].

2 Three View Triangulation

The main motivation for triangulation from more than two views is to use the additional information to improve accuracy. In this section we briefly outline the approach we take and derive the equations to be used in the following sections. This part is essentially identical to that used in [6]. 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 3×4 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. In standard coordinates, this can be written as

$$x_i = \frac{1}{P_{i3}X} \begin{bmatrix} P_{i1}X \\ P_{i2}X \end{bmatrix}, \tag{1}$$

where e.g. P_{i3} refers to row 3 of camera i.

As mentioned previously, we aim at minimizing 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. With this choice of coordinates, we obtain the following cost function to minimize over X

$$\varphi(X) = \frac{(P_{11}X)^2 + (P_{12}X)^2}{(P_{13}X)^2} + \frac{(P_{21}X)^2 + (P_{22}X)^2}{(P_{23}X)^2} + \frac{(P_{31}X)^2 + (P_{32}X)^2}{(P_{33}X)^2}. \tag{2}$$

The approach we take is based on calculating the complete set of stationary points of $\varphi(X)$, i.e. solving $\nabla\varphi(X) = 0$. By inspection of (2) we see that $\nabla\varphi(X)$ will be a sum of rational functions. 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 of $X = [\, X_1 \; X_2 \; X_3 \,]$. To simplify the equations we also make a change of world coordinates, setting the last rows of the respective cameras to

$$P_{13} = [\, 1\; 0\; 0\; 0 \,],\; P_{23} = [\, 0\; 1\; 0\; 0 \,],\; P_{33} = [\, 0\; 0\; 1\; 0 \,]. \tag{3}$$

Since we multiply with the denominator we introduce new stationary points in our equations corresponding to one of the denominators in (2) being equal to zero. This happens precisely when X coincides with the plane through one of the focal points parallel to the corresponding image plane. Such points have infinite/undefined value of $\varphi(X)$ and can therefore safely be removed.

To summarise, we now have three sixth degree equations in three unknowns. The remainder of the theoretical part of the paper will be devoted to the problem of solving these.

3 Using Gröbner Bases To Solve Polynomial Equations

In this section we give an outline of how Gröbner basis techniques can be used for solving systems of multivariate polynomial equations. Gröbner bases are a

concept within algebraic geometry, which is the general theory of multivariate polynomials over any field. Naturally, we are only interested in real solutions, but since algebraic closedness is important to the approach we take, we seek solutions in $\mathbb{C}$ and then ignore any complex solutions we obtain. See e.g. [8] for a good introduction to algebraic geometry.

Our 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 and is a generalization of the companion matrix for one-variable polynomials. From algebraic geometry it is known that the zeros of the equation system can be obtained from the eigenvectors/eigenvalues of the action matrix just as the eigenvectors/eigenvalues of the companion matrix yields the zeros of a one-variable polynomial [9]. The solutions can be extracted from the eigenvalue decomposition in a few different ways, but easiest is perhaps to use the fact that the vector of monomials spanning $\mathbb{C}[\mathbf{x}]/I$ evaluated at a zero of I is an eigenvector of $\mathbf{m}_{x_k}^t$. An alternative is to use the eigenvalues of $\mathbf{m}_{x_k}^t$ corresponding to the values of x_k at zeros of I.

$\mathbb{C}[\mathbf{x}]/I$ 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, unique representative for each equivalence class. Our main focus is therefore on how to compute this Gröbner basis in an efficient and reliable way.

4 Numerical Gröbner Basis Computation

There is a general method for constructing a Gröbner basis known as *Buchberger's algorithm* [9]. It is a generalization of the Euclidean algorithm for computing the greatest common divisor and Gaussian elimination. The general idea is to arrange all monomials according to some ordering and then succesively eliminate leading monomials from the equations in a fashion similar to how Gaussian elimination works. This is done by selecting polynomials pair-wise 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 [10]. 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[2] 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

$$\mathbf{C}\boldsymbol{\varphi} = 0, \tag{4}$$

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

5 Constructing a Gröbner Basis For the Three View Triangulation Problem

As detailed in Section 2, we optimize the L_2 cost function by calculation of the stationary points. This yields three sixth degree polynomial equations in $X = [X_1, X_2, X_3]$. In addition to this, we add a fourth equation by taking the sum of our three original equations. This cancels out the leading terms, producing a fifth degree equation which will be useful in the subsequent calculations [6]. These equations generate an ideal I in $\mathbb{C}[X]$. The purpose of this section is to give the details of how a Gröbner basis for I can be constructed.

First, however, we need to deal with the problem where one or more of $X_i = 0$. When this happens, we get a parametric solution to our equations. As mentioned in Section 2, this corresponds to the extra stationary points introduced by multiplying up denominators and these points have infinite value of the cost function $\varphi(X)$. Hence, we would like to exclude solutions with any $X_i = 0$ or equivalently $X_1X_2X_3 = 0$. The algebraic geometry way of doing this is to calculate the *saturation* $\mathrm{sat}(I, X_1X_2X_3)$ of I w.r.t. $X_1X_2X_3$, consisting of all polynomials $f(X)$ s.t. $(X_1X_2X_3)^k \cdot f \in I$ for some k.

Computationally it is easier to calculate $\mathrm{sat}(I, X_i)$ for one variable at a time and then joining the result. This removes the same problematic parameter family of solutions, but with the side effect of producing some extra (finite) solutions with $X_i = 0$. These do not present any serious difficulties though since they can easily be detected and filtered out.

Consider one of the variables, say X_1. The ideal $\mathrm{sat}(I, X_1)$ is calculated in three steps. We order the monomials according to X_1 but take the monomial with the highest power of X_1 to be the smallest, e.g. $X_1X_2^2X_3 \geq X_1^2X_2^2X_3$. With the monomials ordered this way, we perform a few steps of the Gröbner basis

[2] Usually with the aid of some algebraic geometry software as Macaulay 2 [11]

calculation, yielding a set of generators where the last elements can be divided by powers of X_1. We add these new equations which are "stripped" from powers of X_1 to I.

More concretely, we multiply the equations by all monomials creating equations up to degree seven. After the elimination step two equations are divisible by X_1 and one is divisible by X_1^2.

The saturation process is performed analogously for X_2 and X_3 producing the saturated ideal I_{sat}, from which we extract our solutions.

The final step is to calculate a Gröbner basis for I_{sat}, at this point generated by a set of nine fifth and sixth degree equations. To be able to do this we multiply with monomials creating 225 equations in 209 different monomials of total degree up to nine (refer to [6] for more details on the saturation and expansion process outlined above). The last step thus consists of putting the 225 by 209 matrix $\mathbf{C}$ on reduced row echelon form.

This last part turns out to be a delicate task though 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 [6]. In the next section, we propose a strategy for dealing with this problem which drastically improves numerical precision allowing us to use standard IEEE double precision.

6 The Relaxed Ideal Method

After the saturation step, we have a set of equations which "tightly" describe the set of solutions and nothing more. It turns out that by relaxing the constraints somewhat, possibly allowing some extra spurious solutions to enter the equations, we get a significantly better conditioned problem. We thus aim at 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. Forming the variety/solution set V of an ideal is an inclusion reversing operation and hence we have $V(I) \subset V(I_{\text{rel}})$, which means that we are guaranteed not to lose any solutions. Moreover, since all monomials of degree nine are represented in exactly one of our generators for I_{rel}, this means that by construction we have a Gröbner basis for I_{rel}. 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.

The claim that the number of solutions is equal to the dimension of $\mathbb{C}[X]/I$ only holds if I is a radical ideal. Otherwise, the dimension is only an *upper bound* on the number of solutions [8]. Furthermore, as mentioned in Section 3, a

necessary condition for a specific point to be a solution is that the vector of basis monomials evaluated at that point is an eigenvector of the transposed action matrix. This condition is however not sufficient and there can be eigenvectors that do not correspond to zeros of the ideal. This will be the case if I is not a radical ideal. This can lead to false solutions, but does not present any serious problems since false solutions can easily be detected by e.g. evaluation of the original equations.

Since we have 55 leading monomials in the Gröbner basis, the 154 remaining monomials (of the totally 209 monomials) form a basis for $\mathbb{C}/I_{\text{rel}}$. Since I_{rel} was constructed from our original equations by multiplication with monomials and invertible row operations (by Q^t) we expect there to be no new actual solutions. This has been confirmed empirically.

One can therefore say that starting out with a radical ideal I, we relax the radicality property and compute a Gröbner basis for a non-radical ideal but with the same set of solutions. This way we improve 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.

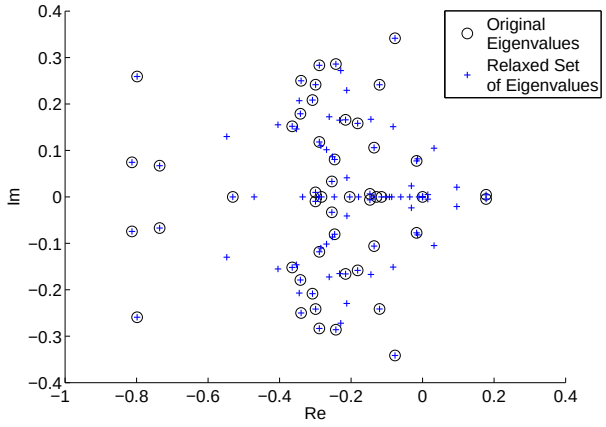

Fig. 1. Eigenvalues of the action matrix using the standard method and the relaxed ideal method respectively, plotted in the complex number plane. The latter are a strict superset of the former.

7 Experimental Validation

The algorithm described in this paper has been implemented in C++ making use of optimized LAPACK and BLAS implementations [12] and the code is

available for download from [7]. The purpose of this section is to evaluate the algorithm in terms of speed and numerical precision. 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.* [6], which needs 30 seconds per triangulation with their setup. The branch and bound method of [5] is faster than [6] but exact running times for triangulation are not given in [5]. However, based on the performance of this algorithm on similar problems, the running time for three view triangulation is probably at least a couple of seconds using their method.

7.1 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 [6] implemented in double precision here referred to as the standard method since it is based on straightforward Gröbner basis calculation. 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.

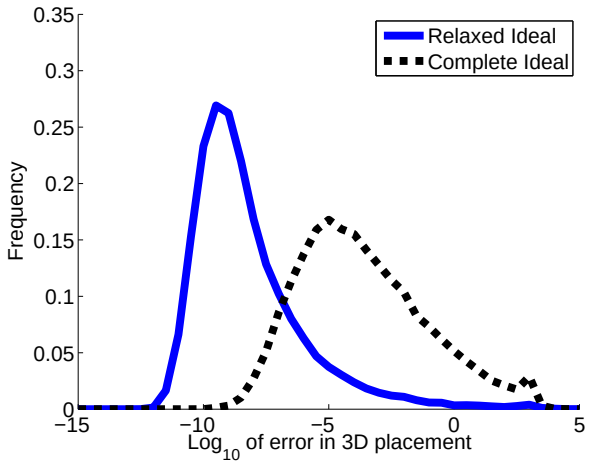

Fig. 2. Histogram over the error in 3D location of the estimated point X. As is evident from the graph, extracting solutions from the smaller ideal yields a final result with considerably smaller errors.

Since we consider triangulation by minimization of the L_2 norm of the error, ideally behaviour under noise should not be affected by the algorithm used. In the second experiment we assert that our algorithm behaves as expected under noise. We generate data as in the first experiment and apply gaussian noise to the image measurements in 0.1 pixel intervals from 0 to 5 pixels. We triangulate 1000 points for each noise level. The median error in 3D location is plotted versus noise in Fig. 3. There is a linear relation between noise and error, which confirms that the algorithm is stable also in the presence of noise.

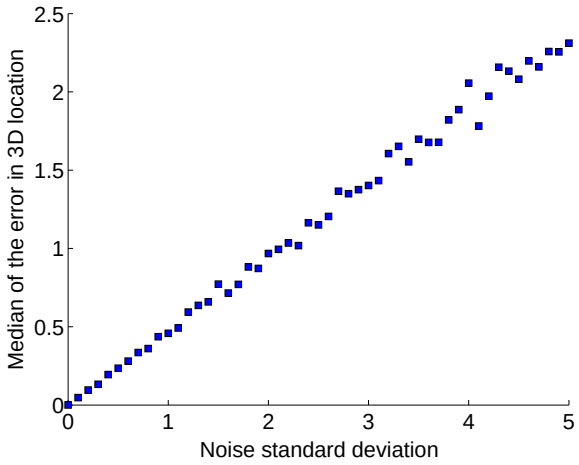

Fig. 3. Error in 3D location of the triangulated point X as a function of image-point noise. The behaviour under noise is as expected given the problem formulation.

7.2 A Real Example

Finally, we evaluate the algorithm under real world conditions. The Oxford dinosaur [13] is a familiar 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. 4.

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 [6] 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.

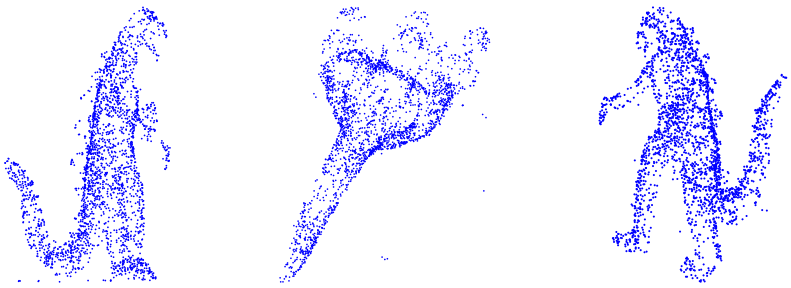

Fig. 4. The Oxford dinosaur reconstructed from 2683 point triplets using the method described in this paper. The reconstruction was completed in approximately 2.5 minutes.

8 Conclusions

In this paper we have shown how a typical problem from computer vision, 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. In all fairness though, linear initialisation combined with bundle adjustment will probably remain the choice for most applications since this is still significantly faster and gives excellent accuracy. However, if a guarantee of finding the provably optimal solution is desired, we provide a competetive method.

More importantly perhaps, 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.

Currently the limiting factor in many applications of Gröbner bases is numerical difficulties. Using the technique presented in this paper of computing the Gröbner basis of a smaller/relaxed ideal, we are able to improve the numerical precision by approximately a factor 10^5. We thus show that there is room for improvement on this point and there is certainly more to explore here. For instance, our choice of relaxation is somewhat arbitrary. Would it be possible to select more/other equations and get better results? If more equations can be kept, but with retained accuracy this is certainly a gain since it allows an eigenvalue decomposition of a smaller action matrix and this operation in most cases has $\mathcal{O}(n^3)$ time complexity.

Acknowledgment

This work has been funded by the Swedish Research Council through grant no. 2005-3230 'Geometry of multi-camera systems', grant no. 2004-4579 'Image-Based Localisation and Recognition of Scenes', SSF project VISCOS II and the European Commission's Sixth Framework Programme under grant no. 011838 as part of the Integrated Project SMErobot.

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
2. Hartley, R., Sturm, P.: Triangulation. Computer Vision and Image Understanding **68** (1997) 146–157
3. Triggs, W., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment: A modern synthesis. In Triggs, W., Zisserman, A., Szeliski, R., eds.: Vision Algorithms: Theory and Practice. LNCS. Springer Verlag (2000)
4. Kahl, F.: Multiple view geometry and the L_∞-norm. In: International Conference on Computer Vision, Beijing, China (2005) 1002–1009
5. Agarwal, S., Chandraker, M.K., Kahl, F., Kriegman, D.J., Belongie, S.: Practical global optimization for multiview geometry. In: Proc. 9th European Conf. on Computer Vision, Graz, Austria. (2006) 592–605
6. Stewénius, H., Schaffalitzky, F., Nistér, D.: How hard is three-view triangulation really? In: Proc. Int. Conf. on Computer Vision, Beijing, China (2005) 686–693
7. : Three view triangulation. (http://www.maths.lth.se/∼byrod/downloads.html)
8. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms. Springer (2007)
9. Cox, D., Little, J., O'Shea, D.: Using Algebraic Geometry. Springer Verlag (1998)
10. Faugère, J.C.: A new efficient algorithm for computing gröbner bases (f_4). Journal of Pure and Applied Algebra **139**(1-3) (1999) 61–88
11. Grayson, D., Stillman, M.: Macaulay 2. (http://www.math.uiuc.edu/Macaulay2)
12. : Lapack - linear algebra package. (http://www.netlib.org/lapack)
13. : Visual geometry group, university of oxford. (http://www.robots.ox.ac.uk/∼vgg)