



# LUND UNIVERSITY

## Camera Resectioning from a Box

Aanæs, Henrik; Josephson, Klas; Anton, François; Bærentzen, J. Andreas; Kahl, Fredrik

*Published in:*  
Lecture Notes in Computer Science

2009

[Link to publication](#)

*Citation for published version (APA):*  
Aanæs, H., Josephson, K., Anton, F., Bærentzen, J. A., & Kahl, F. (2009). Camera Resectioning from a Box. In *Lecture Notes in Computer Science* Springer.

*Total number of authors:*  
5

### General rights

Unless other specific re-use rights are stated the following general rights apply:  
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00



# LUND UNIVERSITY

Centre for Mathematical Sciences

---

# LUP

Lund University Publications  
Institutional Repository of Lund University  
Found at: <http://www.lu.se>

This is an author produced version of a paper presented  
at  
16th Scandinavian Conferences on Image Analysis, June  
15-18, Oslo, Norway

This paper has been peer-reviewed but does not include  
the final publisher proof-corrections or journal  
pagination.

Citation for the published paper:  
Author: Henrik Aanæs, Klas Josephson, Francois Anton,  
J. Andreas Bærentzen and Fredrik Kahl  
Title: Camera Resectioning from a Box,  
Journal: Lecture Notes in Computer Science, 2009

Access to the published version may require  
subscription. Published with permission from:  
Springer Netherlands



# Camera Resectioning from a Box

Henrik Aanaes<sup>1</sup>, Klas Josephson<sup>2</sup>, François Anton<sup>1</sup>, J. Andreas Bærentzen<sup>1</sup>,  
and Fredrik Kahl<sup>2</sup>

<sup>1</sup> DTU Informatics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

<sup>2</sup> Centre For Mathematical Sciences, Lund University, Lund, Sweden

**Abstract.** In this paper we describe how we can do camera resectioning from a box with unknown dimensions, i.e. determine the camera model, assuming that image pixels are square. This assumption is equivalent to assuming that the camera has an aspect ratio of one and zero skew, and this holds for most — if not all — digital cameras. Our proposed method works by first deriving 9 linear constraints on the projective camera matrix from the box, leaving a 3-dimensional subspace in which the projective camera matrix can lie. A single solution in this 3D subspace is then found via a method by Triggs in 1999, which uses the square pixel assumption to set up a 4th degree polynomial to which the solution is the desired model. This approach is, however, numerically challenging, and we use several means to tackle this issue. Lastly the solution is refined in an iterative manner, i.e. using bundle adjustment.

## 1 Introduction

With the ever increasing use of interactive 3D environments for online social interaction, computer gaming and online shopping, there is also an ever increasing need for 3D modelling. And even though there has been a tremendous increase in our ability to process and display such 3D environments, the creation of such 3D content is still mainly a manual — and thus expensive — task. A natural way of automating 3D content creation is via image based methods, where several images are taken of a real world object upon which a 3D model is generated, c.f. e.g. [9,12]. However, such fully automated image based methods do not yet exist for general scenes. Hence, we are contemplating doing such modelling in a semi-automatic fashion, where 3D models are generated from images with a minimum of user input, inspired e.g. by Hengel et al. [18].

For many objects, especially man made, boxes are a natural building blocks. Hence, we are contemplating a system where a user can annotate the bounding box of an object in several images, and from this get a rough estimate of the geometry, see Figure 1. However, we do not envision that the user will supply the dimensions (even relatively) of that box. Hence, in order to get a correspondence between the images, and thereby refine the geometry, we need to be able to do camera resectioning from a box. That is, given an annotation of a box, as seen in Figure 1, we should be able to determine the camera geometry. At present, to the best of our knowledge, no solution is available for this particular resectioning

problem, and such a solution is what we present here. Thus, taking the first step towards building a semi-automatic image based 3D modelling system.

Our proposed method works by first extracting 9 linear constraints from the geometry of the box, as explained in Section 2, and thereupon resolving the ambiguity by enforcing the constraint that the pixels should be square. Our method extends the method of Triggs [16] from points to boxes, does not require elimination of variables, and is numerically more stable. Moreover, the complexity of our method is polynomial by opposition to the complexity of the method of Triggs, which is doubly exponential. It results in solving a 4th degree polynomial system in 2 variables. This is covered in Section 3. There are however some numerical issues which need attention as described in Section 4. Lastly our solution is refined via Bundle adjustment c.f. e.g. [17].



**Fig. 1.** A typical man made object, which at a coarse level is approximated well by a box. It is the annotation of such a box, that we assume the user is going to do in a sequence of images.

### 1.1 Relation to Other Work

Solutions to the camera resectioning problem are by no means novel. For the uncalibrated pinhole camera model the resectioning problem can be solved from 6 or more points via a direct linear transform from 6 or more points c.f. e.g. [9], using so called algebraic methods. If the camera is calibrated, in the sense that the internal parameters are known, solutions exist for 3 or more known 3D points c.f. e.g. [8], given that the camera is a pinhole camera. In the general case – the camera is not assumed to obey the pinhole camera model – of a calibrated

camera and 3 or more points, Nister et al. [11] have provided a solution. In the rest of this paper, a pinhole camera model is assumed. A linear algorithm for resectioning of a calibrated camera from 4 or more points or lines [1] exists. If parts of the intrinsic camera parameters are known, e.g. that the pixels are square, solutions also exist c.f. e.g. [16]. Lastly, we would like to mention that from a decent initial estimate we can solve any – well posed – resection problem via bundle adjustment c.f. e.g. [17].

Most of the methods above require the solution to a system of multivariate polynomials, c.f. [5,6]. And also many of these problems end up being numerically challenging as addressed within a computer vision context in [3].

## 2 Basic Equations

Basically, we want to do camera resectioning from the geometry illustrated in Figure 2, where  $a$  and  $b$  are unknown. The two remaining corners are fixed to  $(0, 0, 0)$  and  $(1, 0, 0)$  in order to fix a frame of reference, and thereby remove the ambiguity over all scale rotations and translations. Assuming a projective or pinhole camera model,  $\mathbf{P}$ , the relationship between a 3D point  $Q_i$  and it's corresponding 2D point  $q_i$  is given by

$$q_i = \mathbf{P}Q_i \text{ ,} \quad (1)$$

where  $Q_i$  and  $q_i$  are in homogeneous coordinates, and  $\mathbf{P}$  is a 3 by 4 matrix. It is known that  $Q_i$  and  $q_i$  induces the following linear constraint on  $\mathbf{P}$ , c.f. [9]

$$0 = [q_i]_x \mathbf{P}Q_i = Q_i^T \otimes [q_i]_x \bar{P} \text{ ,} \quad (2)$$

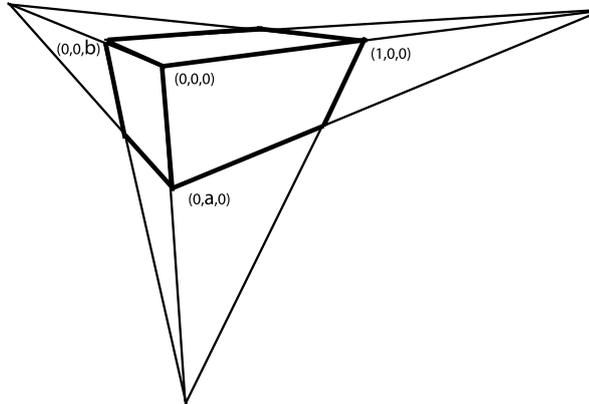
where  $[q_i]_x$  is the 3 by 3 matrix corresponding to taking the cross product with  $q_i$ ,  $\otimes$  is the Kronecker product and  $\bar{P}$  is the elements of  $\mathbf{P}$  arranged as a vector. Setting  $c_i = Q_i^T \otimes [q_i]_x$ , and arranging the  $c_i$  in a matrix  $\mathbf{C} = [c_1^T, \dots, c_n^T]^T$ , we have a linear system of equations

$$\mathbf{C}\bar{P} = 0 \quad (3)$$

constraining  $\mathbf{P}$ . This is the method used here.

To address the issue that we do not know  $a$  and  $b$ , we assume that the box has right angles, in which case the box defines points at infinity. These points at infinity are, as illustrated in Figure 2, independent of the size of  $a$  and  $b$ , and can be derived by calculating the intersections of the lines composing the edges of the box.<sup>3</sup> We thus calculate linear constraints,  $c_i$ , based on  $[0, 0, 0, 1]^T$  and  $[1, 0, 0, 1]^T$  and the three points at infinity  $[1, 0, 0, 0]^T$ ,  $[0, 1, 0, 0]^T$ ,  $[0, 0, 1, 0]^T$ . This, however, only yields 9 constraints on  $\mathbf{P}$ , i.e. the rank of  $\mathbf{C}$  is 9. Usually a 3D to 2D point correspondence gives 2 constraints, and we should have 10 constraints. The points  $[0, 0, 0, 1]^T$ ,  $[1, 0, 0, 1]^T$  and  $[1, 0, 0, 0]^T$  are, however, located on a line making them partly linearly dependant, and thus giving an extra degree of freedom, leaving us with our 9 constraints.

<sup>3</sup> Note that in projective space infinity is a point like any other.



**Fig. 2.** The geometric outline of the box, from which we want to do the resectioning, along with the associated points at infinity denoted. Here  $a$  and  $b$  are the unknowns.

To define  $\mathbf{P}$  completely we need 11 constraints, in that it has 12 parameters and is independent of scale. The null space of  $\mathbf{C}$  is thus (by the dimension theorem for subspaces) 3-dimensional instead of 1-dimensional. We are thus 2 degrees short. By requiring that the images are taken by a digital camera the pixels should be perfectly square. This assumption gives us the remaining two degrees of freedom, in that a pinhole camera model has a parameters for skewness of the pixels as well as one for their aspect ratio. The issue is, however, how to incorporate these two constraints in a computationally feasible way. In order to do this, we will let the 3D right-null space of  $\mathbf{C}$  be spanned by  $v_1, v_2, v_3$ . The usual way to find  $v_1, v_2, v_3$  is via singular value decomposition (SVD) of  $\mathbf{C}$ . But during our experiments we found that it does not yield the desired result. Instead, one of the equations in  $\mathbf{C}$  corresponding to the point  $[0, 0, 0, 1]^T$  was removed, and by that, we can calculate the null space of the remaining nine equations. This turned out to be a crucial step to get the proposed method to work. We have also tried to remove any of the theoretically linearly dependent equations, and the result proved not to be dependent on the equations that were removed. Then,  $\mathbf{P}$  is seen to be a linear combination of  $v_1, v_2, v_3$ , i.e.

$$\bar{P} = \mu_1 v_1 + \mu_2 v_2 + \mu_3 v_3 \quad . \quad (4)$$

For computational reasons, we will set  $\mu_3 = 1$ , and if this turns out to be numerically unstable, we will set one of the other coefficients to one.

### 3 Polynomial Equation

Here we are going to find the solution to (4), by using the method proposed by Triggs in [16]. To do this, we decompose the pinhole camera into intrinsic parameters  $\mathbf{K}$ , rotation  $\mathbf{R}$  and translation  $t$ , such that

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|t] . \quad (5)$$

The dual image of the absolute quadric,  $\omega$  is given by [9,16]

$$\omega = \mathbf{P}\Omega\mathbf{P}^T = \mathbf{K}\mathbf{K}^T , \quad (6)$$

where  $\Omega$  is the absolute dual quadric,

$$\Omega = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} .$$

Here  $\mathbf{P}$  and thus  $\mathbf{K}$  and  $\omega$  are functions of  $\mu = [\mu_1, \mu_2]^T$ . Assuming that the pixels are square is equivalent to  $\mathbf{K}$  having the form

$$\mathbf{K} = \begin{bmatrix} f & 0 & \Delta x \\ 0 & f & \Delta y \\ 0 & 0 & 1 \end{bmatrix} , \quad (7)$$

where  $f$  is the focal length and  $(\Delta x, \Delta y)$  is the optical center of the camera. In this case the the upper 2 by 2 part of  $\omega^{-1}$  is proportional to an identity matrix. Using the matrix of cofactors, it is seen that this corresponds to the minor of  $\omega_{11}$  equals the minor of  $\omega_{22}$  and that the minor of  $\omega_{12}$  equals 0, i.e.

$$\omega_{22}\omega_{33} - \omega_{23}^2 = \omega_{11}\omega_{33} - \omega_{13}^2 \quad (8)$$

$$\omega_{21}\omega_{33} - \omega_{23}\omega_{31} = 0 \quad (9)$$

This corresponds to a fourth degree polynomial in the elements of  $\mu = [\mu_1, \mu_2]^T$ . Solving this polynomial equation will give us the linear combination in (4), corresponding to a camera with square pixels, and thus the solution to our resectioning problem.

#### 3.1 Polynomial Equation Solver

To solve the system of polynomial equations Gröbner basis methods are used. These methods compute the basis of the vector space (called the quotient algebra) of all the unique representatives of the residuals of the (Euclidean) multivariate division of all polynomials by the polynomials of the system to be solved, without relying on elimination of variables, nor performing the doubly exponential time computation of the Gröbner basis. Moreover, this computation of the Gröbner basis, which requires the successive computation of remainders in floating point arithmetic, would induce an explosion of the errors. This approach

has been a successful method used to solve several systems of polynomial equations in computer vision in recent years e.g. [4,13,14]. The pros of Gröbner basis methods is that they give a fast way to solve systems of polynomial equations, and that they reduce the problem of the computation of these solutions to a linear algebra (eigenvalue) problem, which is solvable by radicals if the size of the matrix does not exceed 4, yielding a closed form in such cases. On the other hand the numerical accuracy can be a problem [15]. A simple introduction to Gröbner bases and the field of algebraic geometry (which is the theoretical basis of the Gröbner basis) can be found in the two books by Cox et al. [5,6].

The numerical Gröbner basis methods we are using here require that the number of solutions to the problem needs to be known beforehand, because we do not actually compute the Gröbner basis. An upper bound to a system is given by Bézout's theorem [6]. It states that the number of solutions of a system of polynomial equations is generically the product of the degrees of the polynomials. The upper bound is reached only if the decompositions of the polynomials into irreducible factors do not have any (irreducible) factor in common. In this case, since there are two polynomials of degree four in the system to be solved, the maximal number of solutions is 16. This is also the true number of complex solutions of the problem. The number of solutions is later used when the action (also called the multiplication map in algebraic geometry) matrix is constructed, it is also the size of the minimal eigenvalue problem necessary to solve. We are using a threshold to determine whether monomials are certainly standard monomials (which are the elements of the basis of the quotient algebra) or not. The monomials for which we are not sure whether they are standard are added to the basis, yielding a higher dimensional representation of the quotient algebra.

The first step when a system of polynomial equations is solved with such a numerical Gröbner basis based quotient algebra representation is to put the system in matrix form. A homogenous system can be written,

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

In this equation  $\mathbf{C}$  holds the coefficients in the equations and  $X$  the monomials. The next step is to expand the number of equation. This is done by multiplying the original equations by a handcrafted set of monomials in the unknown variables. This is done to get more linearly independent equations with the same set of solutions. For the problem in this paper we multiply with all monomials up to degree 3 in the two unknown variables  $\mu_1$  and  $\mu_2$ . The result of this is twenty equations with the same solution set as the original two equations. Once again we put this on matrix form,

$$\mathbf{C}_{exp}X_{exp} = 0, \tag{11}$$

in this case  $\mathbf{C}_{exp}$  is a  $20 \times 36$  matrix. From this step the method of [3] is used. By using those methods with truncation and automatic choice of the basis monomials the numeric stability is considerably improved. The only parameters that are left to choose is the variable used to construct the action matrix and the truncation threshold. We choose  $\mu_1$  as action variable and the truncation threshold is fixed to  $10^{-8}$ .

An alternative way to solve the polynomial equation is to use the automatic generator for minimal problems presented by Kukelova et al. [10]. A solver generated this way doesn't use the methods of basis selection, which will reduce the numerical stability. We could also use exact arithmetic for computing the Gröbner basis exactly, but this would yield in the tractable cases a much longer computation time, and in the other cases an aborted computation due to a memory shortage.

### 3.2 Resolving Ambiguity

It should be expected that there are more than one real valued solution to the polynomial equations. To determine which of those solutions are correct, an alternative method to calculate the calibration matrix,  $\mathbf{K}$ , is used. After that, the solution from the polynomial equations with a calibration matrix closest to the alternatively calculated calibration matrix is used. The method used is described in [9]. It uses that in the case of square pixels and zero skew the image of the absolute conic has the form

$$\omega^{-1} = \begin{bmatrix} \omega_1 & 0 & \omega_2 \\ 0 & \omega_1 & \omega_3 \\ \omega_2 & \omega_3 & \omega_4 \end{bmatrix} \quad (12)$$

and that for each pair of orthogonal vanishing points  $v_i, v_j$  the relation  $v_i^T \omega^{-1} v_j = 0$  holds. The three orthogonal vanishing points known from the drawn box in the image thus gives three constraints on  $\omega^{-1}$  that can be expressed on matrix form according to  $\mathbf{A}\bar{\omega}^{-1} = 0$  where  $\mathbf{A}$  is a  $3 \times 4$  matrix. The vector  $\bar{\omega}^{-1}$  can then be found as the null space of  $\mathbf{A}$ . The calibration matrix is then obtained by calculating the Cholesky factorization of  $\omega$  as described in equation 6.

The use of the above method also has an extra advantage. Since it doesn't enforce  $\omega$  to be positive definite it can be used as a method to detect uncertainty in the data. If  $\omega$  isn't positive definite, the Cholesky factorization can't be performed and, hence, the result will not be good in the solution of the polynomial equations. To nevertheless have something to compare with, we substitute  $\omega$  with  $\omega - \delta \mathbf{I}$ , where  $\delta$  equals the smallest eigenvalue of  $\omega$  times 1.1.

To decide which solution from the polynomial equations to use the extra constraints that the two points  $[0, 0, 0]$  and  $[1, 0, 0]$  are in front of the camera is enforced. Among those solutions fulfilling this constraint the solution with smallest difference in matrix norm between the calibration matrix from the method described above and those from the solutions of the polynomial equations is used.

## 4 Numerical Considerations

The most common use of Gröbner basis solvers is in the core of a RANSAC engine[7]. In those cases there is no problem if the numerical errors gets large in a few setups since the problem is calculated for many instances and only the best

is used. In the problem of this paper this is not the case instead we need a good solution for every null space used in the polynomial equation solver. To find the best possible solution the accuracy of the solution is measured by the condition number of the matrix that is inverted when the Gröbner basis is calculated. This has been shown to be a good marker of the quality of the solution [2]. Since the order of the vectors in the null space is independent we choose to try a new ordering if this condition number is larger than  $10^5$ . If all orderings gives a condition number larger than  $10^5$  we choose the solution with the smallest condition number. By this we can eliminate the majority of the large errors.

To even further improve the numerical precision the first step in the calculation is to change the scale of the images. The scale is chosen so that the largest absolute value of any image coordinate of the drawn box equals one. By doing this the condition number of  $\omega$  decreases from approximately  $10^6$  to one for an image of size 1000 by 1000.

## 5 Experimental Results

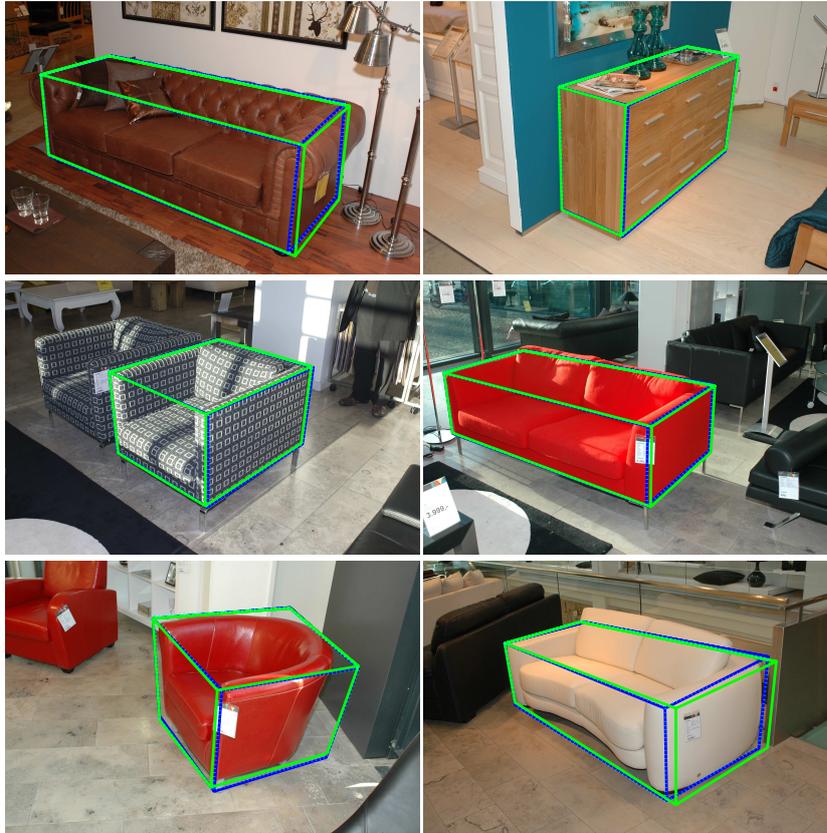
To evaluate the proposed method we went to the local furniture store and took several images of their furniture, e.g. Figure 1. On this data set we manually annotated 30 boxes, outlining furniture, see e.g. Figure 3, and ran our proposed method on the annotated data to get an initial result, and refined the solution with a bundle adjuster. In all but one of these we got acceptable results, in the last example, there were no real solutions to the polynomial equations. As seen from Figure 3, the results are fully satisfactory, and we are now working on using the proposed method in a semi-automatic modelling system. As far as we can see, the reason that we can refine the initial results is that there are numerical inaccuracies in our estimation. To push the point, that fact that we can find a good fit of a box, implies that we have been able to find a model, consisting of camera position and internal parameters as well as values for the unknown box sides  $a$  and  $b$ , that explains the data well. Thus, from the given data, we have a good solution to the camera resectioning problem.

## 6 Conclusion

We have proposed a method for solving the camera resectioning problem from an annotated box, assuming only that the box has right angles, and that the camera's pixels are square. Once several numerical issues have been addressed, the method produces good results.

## Acknowledgements

We wish to thank ILVA A/S in Kgs. Lyngby for helping us gather the furniture images used in this work. This work has been partly funded by the European Research Council (GlobalVision grant no. 209480), the Swedish Research Council



**Fig. 3.** Estimated boxes. The annotated boxes from furniture images denoted blue lines. The initial estimate is denoted by green lines, and the final result is denoted by a dashed magenta line.

(grant no. 2007-6476) and the Swedish Foundation for Strategic Research (SSF) through the programme Future Research Leaders.

## References

1. A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.
2. M. Byröd, K. Josephson, and K. Åström. Improving numerical accuracy of gröbner basis polynomial equation solvers. In *International Conference on Computer Vision*, 2007.
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.

4. M. Byröd, Z. Kukelova, K. Josephson, T. Pajdla, and K. Åström. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. In *Conference on Computer Vision and Pattern Recognition*, 2008.
5. D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer, 2nd edition, 2005.
6. D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, 2007.
7. M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
8. R.M. Haralick, Chung-Nan Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.
9. R. I. Hartley and A. Zisserman. *Multiple View Geometry – 2nd edition*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2003.
10. M. Kukelova, Z. andBujnak and Tomás Pajdla. Automatic generator of minimal problem solvers. In *The 10th European Conference on Computer Vision*, pages 302–315, 2008.
11. D. Nister and H. Stewenius. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision*, 27(1):67–79, 2007.
12. S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 1:519–528, 2006.
13. H. Stewenius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.
14. H. Stewenius, D. Nister, F. Kahl, and F. Schaffilitzky. A minimal solution for relative pose with unknown focal length. *Image and Vision Computing*, 26(7):871–877, 2008.
15. H. Stewenius, 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.
16. B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *In Proc. 7th Int. Conf. on Computer Vision*, pages 278–284. IEEE Computer Society Press, 1999.
17. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Special sessions - bundle adjustment - a modern synthesis. *Lecture Notes in Computer Science*, 1883:298–372, 2000.
18. A. van den Hengel, A. Dick, T. Thormahlen, B. Ward, and P.H.S. Torr. Videotrace: rapid interactive scene modelling from video. *ACM Transactions on Graphics*, 26(3):86–1–5, 2007.