An Algorithm for Self Calibration from Several Views *

Richard I. Hartley
GE - Corporate Research and Development,
P.O. Box 8, Schenectady, NY, 12301.

Abstract

This paper gives a practical algorithm for the self-calibration of a camera from several views. The method involves non-iterative methods for finding an initial calibration for the camera, followed by least-squares iteration to an optimum solution. At the same time, a scaled Euclidean reconstruction of the scene appearing in the images is computed.

1 Introduction

The possibility of calibrating a camera based on the identification of matching points in several views of a scene taken by the same camera has been shown by Maybank and Faugeras ([9, 3]). Using techniques of Projective Geometry they showed that each pair of views of the scene can be used to provide two quadratic equations in the five unknown parameters of the camera. A method of solving these equations to obtain the camera calibration has been reported in [9, 3, 8] based on directly solving these quadratic equations using continuation. It has been reported however that this method requires extreme accuracy of computation, and seems not to be suitable for routine use. In addition with large numbers of cameras (more than three or four) this method threatens to be unworkable.

In this paper a method is given based partly on the well known Levenberg-Marquardt (LM) parameter estimation algorithm ([11]), partly on new non-iterative algorithms and partly on techniques of Projective Geometry for solving this self-calibration problem. This algorithm has the advantage of being applicable to large numbers of views, and in fact performs best when many views are given. As a consequence, the algorithm can be applied to the structure-from-motion problem to determine the structure of a scene from a sequence of views with the same uncalibrated camera. Indeed, since the calibration of the camera may be determined from the correspondence data, it is possible to compute a Euclidean reconstruction of the scene. That is, the scene is reconstructed, relative to the placement of one of the cameras used as reference, up to an unknown scaling.

The algorithm has been demonstrated on real and synthetic data and was shown to perform robustly in the presence of noise.

An extended version of this paper giving more implementation details appears in [6].

*The research described in this paper has been supported by DARPA Contract #MDA972-91-C-0053

2 The Camera Model

A commonly used model for perspective cameras is that of projective mapping from 3D projective space, $\mathcal{P}^3$, to 2D projective space, $\mathcal{P}^2$. This map may be represented by a 3×4 matrix, M of rank 3. The mapping from $\mathcal{P}^3$ to $\mathcal{P}^2$ takes the point $\mathbf{x} = (x, y, z, 1)^\top$ to $\mathbf{u} = M\mathbf{x}$ in homogeneous coordinates.

The matrix M may be decomposed as $M = K(R|-R\mathbf{t})$, where $\mathbf{t}$ represents the location of the camera, R is a rotation matrix representing the orientation of the camera with respect to an absolute coordinate frame, and K is an upper triangular matrix called the *calibration matrix* of the camera.

The entries of the matrix K may be identified with certain physically meaningful quantities known as internal camera parameters.

3 The Euclidean Reconstruction Problem

Consider a situation in which a set of 3D points $\mathbf{x}_j$ are viewed by a set of $N \geq 3$ cameras with matrices M_i numbered from 0 to $N-1$. Denote by $\mathbf{u}^i_j$ the coordinates of the j-th point as seen by the i-th camera. Given the set of coordinates $\mathbf{u}^i_j$ it is required to find the set of camera matrices, M_i and the points $\mathbf{x}_j$. This is the reconstruction problem. A *reconstruction* based on a set of image correspondences $\{\mathbf{u}^i_j\}$ consists of a set of camera matrices M_i and points $\mathbf{x}_j$ such that $M_i\mathbf{x}_j \approx \mathbf{u}^i_j$. (The notation $\approx$ denotes equality up to a non-zero scale factor.) Without further restriction on the M_i or $\mathbf{x}_j$, such a reconstruction is not unique, and may differ by an arbitrary 3D projective transformation from the *true* reconstruction ([2, 4]). Such a reconstruction is called a projective reconstruction. A reconstruction that is known to differ from the true reconstruction by at most a 3D affine transformation is called an affine reconstruction, and one that differs by a Euclidean transformation from the true reconstruction is called a Euclidean reconstruction. The term Euclidean transformation will be used in this paper to mean a similarity transform, namely the composition of a rotation, a translation and a uniform scaling.

In this paper we seek a reconstruction such that all cameras have the same calibration, so that $M_i = K(R_i \mid -R_i\mathbf{t}_i)$, where each R_i is a rotation matrix and K is an upper-triangular matrix, the common calibration matrix of all the cameras. For convenience, it may be assumed further that $R_0 = I$ and $\mathbf{t}_0 = \mathbf{0}$. According to [9] the calibration matrix K is determined by these conditions. Therefore, we are reduced to the problem of reconstruction from views with calibrated cameras. It is well known ([7]) that the relative placement of the cameras may then be computed, up to an indeterminate global scale. Furthermore, the scene may be constructed uniquely with relative to the camera placement. Thus, a Euclidean reconstruction is possible.

4 Direct Iterative Reconstruction

One approach to the Euclidean reconstruction problem is to solve directly for the unknown camera matrices, $M_i = K(R_i \mid -R_i\mathbf{t}_i)$ and points $\mathbf{x}_j$. In particular, we search for M_i of the required form, and $\mathbf{x}_j$ such that $\hat{\mathbf{u}}^i_j = M_i\mathbf{x}_j$ and such that the squared error sum

$$\sum_{i,j} d(\hat{\mathbf{u}}^i_j, \mathbf{u}^i_j)^2 \quad (1)$$

is minimized, where $d(*,*)$ represents Euclidean distance.

This problem may be described in general terms as follows. Given a hypothesized functional relation $\mathbf{Y} = f(\mathbf{X})$ where $\mathbf{X}$ and $\mathbf{Y}$ are vectors in some Euclidean spaces R^m and R^n, and a measured value $\hat{\mathbf{Y}}$ for $\mathbf{Y}$, we wish to find the vector $\hat{\mathbf{X}}$ that most nearly satisfies this functional relation. More precisely, we seek the vector $\hat{\mathbf{X}}$ satisfying $\hat{\mathbf{Y}} = f(\hat{\mathbf{X}}) + \hat{\boldsymbol{\epsilon}}$ for which $||\hat{\boldsymbol{\epsilon}}||$ is minimized. For the Euclidean reconstruction problem the variables $\mathbf{X}$ comprise the 3D coordinates of each of the points $\mathbf{x}$ in space, the rotations R_i of each of the cameras and the common calibration matrix K. The dependent variables $\mathbf{Y}$ comprise the image coordinates $\mathbf{u}^i_j$. The Levenberg-Marquardt (LM) method ([11]) is a popular method of solving problems of this nature, which has been used with success on a wide range of problems. Starting with an initial estimate $\mathbf{X}_0$ it proceeds by iteration to the final solution.

Using the LM method to solve the Euclidean reconstruction problem works well *provided the initial estimate is sufficiently close*. With arbitrary or random guesses at initial values of the parameters it usually fails dismally.

5 Projective Reconstruction

Instead of attempting a direct reconstruction, calibration and pose estimation as in the previous section, we use a two-step approach. In the first step, a reconstruction of the scene is computed, dropping the assumption that the images are all taken with the same camera. The scene configuration obtained in this manner will differ from the true configuration by an unknown 3D projective transformation ([2, 4]). In the second step, this projective transform is estimated. The advantage of proceeding in this manner is that projective reconstruction is relatively straightforward. Then step two, the estimation of the correct 3D transformation, comes down to solving an 8-parameter estimation problem, which is far more tractable than the original problem.

Various methods of projective reconstruction from two or more views have been given previously ([2, 4, 10]). The method given in [4] is a straight-forward non-iterative construction method from two views. Where high precision is required, it should be followed by iterative refinement. Mohr et. al. ([10]) have reported a direct LM approach to projective reconstruction. A different approach using LM has been reported in [6]. This method finds a set of camera matrix M_i and points $\mathbf{x}_i$ such that $M_0 = (I \mid 0)$ to minimize the goal function (1). A linear method ([4]) is used to provide an initial estimate.

6 Converting Projective to Euclidean Reconstruction

Once we have a projective reconstruction of the imaging geometry any other reconstruction (including a desired Euclidean reconstruction) may be obtained by applying a 3D projective transformation. In particular, if $(\{M_i\}, \{\mathbf{x}_j\})$ is a projective reconstruction, then any other reconstruction is of the form $(\{M_iH^{-1}\}, \{H\mathbf{x}_j\})$ where H is a 4×4 non-singular matrix. We seek such a matrix H such that the transformed camera matrices M_iH^{-1} all have the same (yet to be determined) calibration matrix, K. In other words, we seek H such that $M_iH^{-1} = K(R_i \mid -R_i\mathbf{t}_i)$ for all i, where each R_i is a rotation matrix and K is the common upper-triangular calibration matrix.

Without loss of generality, we may make the additional restriction that the zeroeth camera remains located at the origin and that R_0 is the identity. Since in the original projective reconstruction $M_0 = (I \mid 0)$, it follows that H^{-1} may be assumed to have the restricted form

$$H^{-1} = \begin{pmatrix} K & 0 \\ -\mathbf{v}^\top K & \alpha \end{pmatrix} = \begin{pmatrix} I & \mathbf{0} \\ -\mathbf{v}^\top & \alpha \end{pmatrix} \begin{pmatrix} K & 0 \\ 0 & 1 \end{pmatrix} \quad (2)$$

Since the constant α represents scaling in 3-space, we may further assume that $\alpha = 1$. Now, writing each $M_i = (A_i \mid -A_i\mathbf{t}_i)$ and multiplying out leads to a requirement that

$$A_i(I + \mathbf{t}_i\mathbf{v}^\top)K \approx KR_i \quad (3)$$

for some rotation matrix R_i. Our goal is to find K and $\mathbf{v}$ to satisfy this set of conditions. Recall that K is upper triangular, and we may further assume that K_{33} equals 1, hence K contains five unknown entries. The vector $\mathbf{v}$ has a further three unknown entries. In total, it is required to estimate these eight unknown parameters.

Of course, for inexact data, the equations (3) will not be satisfied exactly, and so we will cast this problem as a least-squares minimization problem that may be solved using LM. In particular, given values for K and $\mathbf{v}$, we compute the expression $A_i(I + \mathbf{t}_i\mathbf{v}^\top)K$ for each i (remembering that A_i and $\mathbf{t}_i$ are known). Taking the QR decomposition of this matrix, we obtain upper-triangular matrices K'_i such that

$$A_i(I + \mathbf{t}_i\mathbf{v}^\top)K = K'_i R_i \ . \quad (4)$$

Subsequently, we compute the matrices $X_i = K^{-1}K'_i$ for all i. Since we have assumed that $M_0 = (A_0 \mid -A_0\mathbf{t}_0) = (I \mid 0)$, it follows that $X_0 = I$. Furthermore, if K and $\mathbf{v}$ satisfy the desired condition (3) then $K'_i \approx K$ for all $i > 0$, and so $X_i \approx I$. Accordingly, we seek to minimize the extent by which X_i differs from the identity matrix. Consequently, we multiply each X_i by a normalizing factor α_i chosen so that the sum of squares of diagonal entries of $\alpha_i X_i$ equals 3, and so that $\det \alpha_i X_i > 0$. Now, we seek K and $\mathbf{v}$ to minimize the expression

$$\sum_{i>0} ||\alpha_i X_i - I||^2 \quad (5)$$

Note that each $\alpha_i X_i - I$ is an upper-triangular matrix. This minimization problem fits the general form of LM estimation of a fuction $f : R^8 \mapsto R^{6(N-1)}$ where N is the total number of cameras. The function f maps the eight [1] variable entries of K and $\mathbf{v}$ to the diagonal and above-diagonal entries of $\alpha_i X_i - I$ for $i > 0$. Since this minimization problem involves the estimation of 8 parameters only, it is obviously a great improvement over the original problem as stated in Section 3 that required the simultaneous estimation of the matrix K, the $N-1$ rotation matrices R_i for $i > 0$ and the 3D point coordinates of all points $\mathbf{x}_j$.

It turns out still to be impractical to solve this minimization problem without a good initial guess at K and $\mathbf{v}$. It is possible to take a good prior guess at K if some knowledge of the camera is available. On the other hand, it is difficult to guess the vector $\mathbf{v}$, so it will be necessary to find some way to obtain an initial estimate for $\mathbf{v}$. It will turn out that if $\mathbf{v}$ is known, then the calibration matrix K can be computed by a straight-forward non-iterative algorithm, so there is no need to guess K.

[1] It is possible to assume certain restrictions on the entries of K, such as that skew is zero and that the pixels are square, thereby diminishing the number of variable parameters

7 Euclidean From Affine Reconstruction

With H^{-1} of the form (2) with $\alpha = 1$, the matrix H may be written as

$$H = \begin{pmatrix} K^{-1} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ \mathbf{v}^\top & 1 \end{pmatrix} \ .$$

The right-hand one of these two matrices represents a transformation that moves the plane at infinity, whereas the second one is an affine transformation, not moving the plane at infinity. In fact, if $\mathbf{x}$ is a point being mapped to infinity by the transformation H, then $(\mathbf{v}^\top 1)\mathbf{x} = 0$. So $(\mathbf{v}^\top 1)$ represents the plane that is mapped to the plane at infinity by H.

We will now suppose that by some magic we have been able to determine $\mathbf{v}$. This means, in effect that we know the position of the plane at infinity in the reconstruction. Otherwise stated, we have been able to determine the structure up to an affine transformation. We will now present a simple non-iterative algorithm for the determination of K, and hence of the Euclidean structure.

Equation (3) may be written as $B_i K = KR_i$ where $B_i = \alpha_i A_i(I + \mathbf{t}_i\mathbf{v}^\top)$, and the constant factor α_i is chosen so that $\det B_i = 1$. Matrix B_i is known since A_i, $\mathbf{t}_i$ and $\mathbf{v}$ are assumed known. Consequently, $K^{-1}B_i K = R_i$ is a rotation matrix. Equating, $K^{-1}B_i K = R_i$ with its inverse transpose and rearranging leads to

$$(KK^\top)B_i{}^{-\top} = B_i(KK^\top) \quad (6)$$

where $B_i{}^{-\top}$ is the inverse transpose of B_i. Given sufficiently many views and corresponding matrices B_i, equation 6 may be used to solve for the entries of the matrix $KK^\top$. In particular we denote $KK^\top$ by C, which is a symmetric matrix. Then the equation (6) gives rise to a set of nine linear equations in the six independent entries of C. The matrix C can only be determined up to a constant factor. Because of redundancy, the nine equations derived from (6) for a single known transformation B_i are not sufficient to solve for C. However, if two or more such B_i are known, then we may solve for C.

Once $C = KK^\top$ is found it is an easy matter to solve for K using the Choleski factorization ([1, 11]). A solution for K is only possible when C is positive-definite. This is guaranteed for noise-free data, since by construction, C possesses such a factorization. In cases where the input data is defective, or the plane at infinity is not accurately known it is possible that the matrix C turns out not to be positive-definite, and so the calibration matrix can not be found. In practice however, the algorithm works extremely well, provided the plane at infinity is accurately placed and there are no gross inaccuracies (mistaken matched points) in the data.

8 Quasi-affine Reconstruction

We are interested, however, in finding the plane at infinity without any extra given information. The first step will be to get an approximation to the plane at infinity. This will be done by considering the *cheirality* of the images, in other words, by taking into account the fact that the points must lie in front of the cameras that view them.

The subject of cheirality of cameras was considered in detail in [5]. It was shown in that paper that if $(\{M_i\}, \{\mathbf{x}_j\})$ is a projective reconstruction of a set of image correspondences derived from a real scene, then there exist constants η_j and ϵ_i equal to ± 1, such that $\epsilon_i \eta_j M_i \mathbf{x}_j = (u^i_j, v^i_j, w^i_j)^\top$ where each $w^i_j > 0$. It should be noted that the equality sign here means exact equality, and not equality up to a constant factor. Given the reconstruction $(\{M_i\}, \{\mathbf{x}_j\})$ we may replace M_i by $\epsilon_i M_i$ and $\mathbf{x}_j$ by $\eta_j \mathbf{x}_j$ to obtain a reconstruction such that $M_i \mathbf{x}_j = (u^i_j, v^i_j, w^i_j)^\top$ and each $w^i_j > 0$. Suppose that this has been done. Now ([5]) there exists a matrix $H = \begin{pmatrix} \beta I & 0 \\ \alpha \mathbf{v}^\top & \alpha \end{pmatrix}$ with $\alpha, \beta = \pm 1$ such that $H\mathbf{x}_j = (x'_j, y'_j, z'_j, s'_j)^\top$ with $s'_j > 0$ for all j, and such that $M_i H^{-1} = (A'_i \mid -A'_i \mathbf{t}_i)$ with $\det A'_i > 0$ for all i.

The conditions satisfied by the matrix H transform into inequalities. In particular, $s'_j > 0$ means that

$$\alpha(\mathbf{v}^\top 1)\mathbf{x}_j > 0 \tag{7}$$

for each point $\mathbf{x}_j$. The condition $\det A'_i < 0$ also gives rise to a linear inequality as follows. Writing $M_i = (A_i \mid -A_i \mathbf{t}_i)$ then $M_i H^{-1} = (A'_i \mid -A'_i \mathbf{t}'_i)$ where $A'_i = \beta A_i (I + \mathbf{t}_i \mathbf{v}^\top)$. Then

$$\det A'_i = \beta \det A_i \det(I + \mathbf{t}_i \mathbf{v}^\top) = \beta(1 + {\mathbf{t}_i}^\top \mathbf{v}) \det A_i \ .$$

Since A_i and $\mathbf{t}_i$ are known this gives a linear inequality

$$\beta(1 + {\mathbf{t}_i}^\top \mathbf{v}) \det A_i > 0 \tag{8}$$

in the entries of $\mathbf{v}$. These set of inequalities (7) and (8) constraining the placement of the plane at infinity are called the *cheiral inequalities*.

Naturally, we propose to solve the cheiral inequalities using linear programming (LP). The four cases corresponding to the choices of α and β must be considered. In order to obtain a single solution it is necessary to define an appropriate goal function to optimize. We choose to maximize the margin by which the given inequalities are satisfied, since this should correspond informally to a placement of the plane at infinity at a maximum distance from the points and the cameras. For this to make sense, the homogeneous coordinate expression for $\mathbf{x}_j = (x_j, y_j, z_j, s_j)^\top$ should first be normalized so that $||\mathbf{x}_j|| = 1$. Now, we have a set of inequalities of the form ${\mathbf{f}_i}^\top \mathbf{v} \geq g_i$, where $\mathbf{f}_i$ is simply the vector of coefficients of the i-th equation. We add an extra variable δ to obtain equations of the form ${\mathbf{f}_i}^\top \mathbf{v} - \delta \geq g_i$. The LP problem is to maximize δ subject to the given inequalities. If $\delta > 0$ in the optimum solution, then the original inequalities have a solution, and this is the solution that we accept to obtain $\mathbf{v}$. Once $\mathbf{v}$ has been found by solving the LP problem, the projective reconstruction is transformed by the corresponding matrix H. The new reconstruction may be termed a *quasi-affine* reconstruction.

By solving this cheiral inequalities, we find a candidate value for $\mathbf{v}$. By the method of Section 7 we can now compute the corresponding value of K. This estimate may then be refined using the method described in Section 6. There is one flaw in this scheme, namely that it may not be possible to find K corresponding to the estimated $\mathbf{v}$, because the matrix C, which should equal $KK^\top$, is not positive definite. In this case, it is necessary to select a different $\mathbf{v}$. This may be done by carrying out a random search over the convex region of 3-space defined by the cheirality inequalities. In fact, a reasonable approach is to find several candidate vectors $\mathbf{v}$ and iterate from each of them, finally selecting the best solution. This is what I have done in practice.

9 Algorithm Outline

Since the details of the outline have been obscured by the necessary mathematical analysis, the complete algorithm for Euclidean reconstruction will now be given. To understand the details of the steps of the algorithm, the reader must refer to the relevant section of the previous text.

1. Compute a projective reconstruction of the scene (Section 5)

2. Compute a quasi-affine reconstruction of the scene (Section 8)

3. Search for a quasi-affine reconstruction from which the calibration matrix K may be computed (Section 8)

   (a) For a randomly selected set of vectors $\mathbf{v}$ contained within the region determined by the cheiral inequalities solve the equations $C{B_i}^{-\top} = B_i C$ as described in Section 7 until we find a $\mathbf{v}$ such that the solution C is positive-definite.

   (b) Determine K by Choleski factorization of $C = KK^{\top}$.

4. Carry out LM iteration using the method of Section 6 to find a Euclidean reconstruction.

5. Using the values of K, R_i and $\mathbf{x}_j$ that come out of the previous step, do a complete LM iteration to find the optimal solution minimizing the image-coordinate error, using the method described in Section 4.

Various comments are in order here. First of all, some of the steps in this algorithm may not be necessary. The second step (determination of a specific quasi-affine reconstruction) may not be necessary, since the third step does a search for a modified quasi-affine reconstruction. However, it is included, since it provides a point of reference for the subsequent search. The vector $\mathbf{v}$ found in the third step of the algorithm should be small, so that the modified quasi-affine reconstruction is close to the original one. In fact, as mentioned previously it is possible to use the cheiral inequalities to give bounds on the individual entries in the vector $\mathbf{v}$. Finally, it has been found that the last step of the algorithm, the final iteration is scarcely necessary, and does not make a very large difference to the solution. It commonly decreases the value of the image coordinate error by no more that about 10%, at least when there are many views. In addition, this last step is relatively costly in terms of computation time.

10 Experimental Evaluation

Synthetic Data. Two experiments were carried out with synthetic data – one with three views and one with 15 views. With just three views (the minimum possible) the algorithm's performance was mediocre. For RMS noise levels less than 0.5 pixels in each axial direction in a (700×600 image), the focal length of the camera was accurate within about 10%, but the principal point was displaced by about 250 pixels.

For 15 views, however the algorithm performed extremely well. For an RMS noise level of 8 pixels in each axial direction (a rather high value) the principal point was located within 20 pixels and the two axial scaling factors were accurate to within 2%. The RMS reconstruction error was only 1.45cm for the scene consisting of 50 random points in a 1m radius sphere. Even for a noise level of 16 pixels the reconstruction error was only 3.3cm, whereas for a noise level of 1 pixel (a realistic value), the reconstruction was accurate to within 1.6mm, and the focal lengths were accurate to within 0.1 %.

More details of these experiments are given in [6].

Real Data. The algorithm was also evaluated on a set of image coordinate correspondences kindly supplied by Boubakeur Boufama and Roger Mohr. The object in question was a wooden house, for which 9 views were used and a total of 73 points were tracked. This is the same image set as used in the paper [10]. Figure 1 shows one of the views of the house. The algorithm converged very successfully on this data. The measured residual RMS pixel error was found to be 0.6 pixels per point, which is about as good as can be expected, since the image correspondences were not supplied with sub-pixel accuracy. Reconstruction accuracy, however, could not be tested, and may be suspect. Figure 2 shows a reconstructed view of the set of 73 points looking directly down the edge of the house. Clearly visible is the corner of the house, showing a right-angled corner. This gives evidence for the success of Euclidean reconstruction, since angles are a Euclidean attribute of the scene.

11 Conclusions

The experience gained with the implementation of this algorithm shows that special care is needed in camera calibration using only image correspondences in multiple views. Nevertheless, a multi-step approach to reconstruction, proceeding via a preliminary projective reconstruction can give good results. If only 3 views are given, then the performance of the algorithm is not entirely satisfactory, mainly because the problem is inherently unstable. For larger numbers of views, however, the algorithm works well. This suggests that it may be successfully applied to video sequences taken with a moving camera, or of a moving object.

Figure 1: One of the views of wooden houses

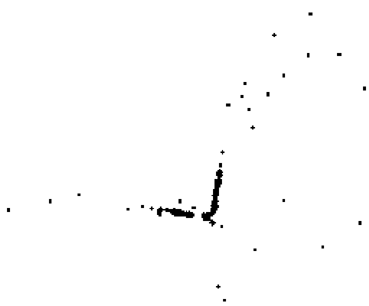

Figure 2: View of reconstructed point set

References

[1] K.E. Atkinson. *An Introduction to Numerical Analysis, 2nd Edition.* John Wiley and Sons, New York, 1989.

[2] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Computer Vision - ECCV '92, LNCS-Series Vol. 588, Springer-Verlag*, pages 563 – 578, 1992.

[3] O. D. Faugeras, Q.-T Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *Computer Vision - ECCV '92, LNCS-Series Vol. 588, Springer-Verlag*, pages 321 – 334, 1992.

[4] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 761–764, 1992.

[5] R. I. Hartley. Cheirality invariants. In *Proc. DARPA Image Understanding Workshop*, pages 745 – 753, 1993.

[6] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proc. of the Second Europe-US Workshop on Invariance, Ponta Delgada, Azores*, pages 187–202, October 1993.

[7] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Sept 1981.

[8] Q.-T Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement.* PhD thesis, Universite de Paris-Sud, Centre D'Orsay, 1992.

[9] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8:2:123 – 151, 1992.

[10] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 543 – 548, 1993.

[11] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1988.