Motion and 3D structure recovery from uncalibrated images

Boubakeur S. Boufama
DEPT. OF MATH AND COMPUTER SCIENCE
University of Prince Edward Island
550 University Avenue, Charlottetown,
PEI Canada C1A 4P3
E-mail : bboufama@upei.ca

Abstract

This paper addresses the problem of computing the camera motion and the Euclidean 3D structure of an observed scene using uncalibrated images. Given at least 2 images with pixel correspondences, the motion of the camera and the 3D structure of the scene are calculated simultaneously. We do not assume the knowledge of the intrinsic parameters of the camera. However, an approximation of these parameters is required. Classical methods based on the essential matrix computation have proven to be very unstable when the intrinsic parameters of the cameras are not known exactly. To overcome such unstability, we propose here a method where a particular choice of a 3D Euclidean coordinate system with a different parameterization of the motion/structure problem allowed us to reduce significantly the total number of unknowns. In addition, the simultaneous calculation of the camera motion and the 3D structure has made the computation of the motion and structure less sensitive to the errors in the values of the intrinsic parameters of the camera. All steps of our method are linear. However, a final nonlinear optimal step might be added to improve the accuracy of the results.

Experiments with real images validated our method and showed that a good quality motion/structure can be recovered from a pair of uncalibrated images.

keywords: Euclidean reconstruction, camera motion, stereovision.

1 Introduction

One of the principal goals of research in computer vision is to enable machines to perceive the three-dimensional nature of the environment. Unfortunately in many cases, the only information we possess about the scene is two-dimensional images. It is well known that recovering depth from a single (two-dimension) image is not possible in a general case. However, if we use more than one image, the problem becomes feasible.

It is possible to reconstruct three-dimensional scenes only from point correspondences in several images [3][8][19]. However, such a reconstruction can only be defined up to a collineation, that is a 3D projective transformation. Because a 3D projective reconstruction lacks metric information, the use of this type of reconstruction is usually limited to some applications such as object recognition (see for instance the book of Rothwell [18]).

Euclidean information is the richest and the most used in computer vision. Hence, the Euclidean 3D reconstruction problem is of central importance in computer vision. The classical way for solving the Euclidean reconstruction problem requires the calibration of the cameras and the matching of the features in the different images. This approach is often nonrealistic. In particular, it is not always possible to calibrate the camera on-line; for example, when the latter is involved in visual tasks. Hence, several researchers have developed methods for obtaining the Euclidean reconstruction without calibrating the camera. Most of these researchers have assumed less general camera models such as the orthographic model [10], affine model [21] or paraperspective model [23]. These methods have their limitations. In particular, they assume that the size of the scene is small compared to the scene-camera distance. Other researchers avoided using a less general camera model, but assumed a less general camera motion, see for instance [16, 22].

A new direction for solving the Euclidean reconstruction has focused on the calibration[1] of the camera. However, unlike the classical calibration method, the new one, called self-calibration, uses only matched points in the images and does not need a calibration pattern. Once the camera is calibrated, the problem of calculating the motion of the camera/scene and the Euclidean reconstruction of the observed scene become straightforward [11]. The possibility of calibrating a camera using only point correspondences in several images of a rigid scene has been shown by Faugeras et al. [5] and, by Maybank and Faugeras [15]. In their paper, the emphasis was placed on the theory of the solution without providing a practical algorithm that is suitable for routine use. The most recent papers on self-calibration, where practical results were shown, are the work of Luong [12] and Hartley [7]. Luong's method is based on Kruppa's equations [9] which express the rigidity constraint of the scene. Each pair of images (or a displacement) provides two quadratic constraints on the 5 intrinsic parameters of the camera[2]. Therefore, at least 2 displacements (3 images) are required to solve for the 5 intrinsic parameters. Because the equations used were quadratic, it has been reported that high accuracy in image points localization and reliable correspondences are necessary and that, not all types of displacements yield stable results. Furthermore, the reconstructions obtained using this self-calibration were not very precise, but still useful for some robotics tasks. Hartley's method uses 3 images of a rigid scene taken from the same point in space; that is, the camera must undergo a pure rotation each time. This self-calibration method has the advantage over Luong's method of being non-iterative. However, unlike Luong's method, each displacement must be a pure rotation. This constraint might be a major drawback in some cases; for instance, when the scene is moving and the camera is not. Furthermore, the author noted that his method works best on wide angle images. Although no comparison has been done between the two methods, it seems that they obtain comparable accuracy for the reconstructions. Both authors seem to agree that the accuracy of their reconstructions compare poorly with the accuracy obtained using the classical calibration methods.

The approach developed in this paper does not aim to challenge any self-calibration method since our goal is reconstruction and not calibration. However, in this paper, we show that good accuracy, at least comparable to the one obtained through self-calibration, on the reconstruction can be achieved. We show, in particular that, unlike self-calibration based methods, our method is linear, simple, practical and requires only a minimun of 2 images. Yet, the reconstruction we obtain is not less accurate than the one obtained by the self-calibration based methods and it is sufficient for most robotics applications.

Section 2 describes the background used by our reconstruction method. Section 2 gives the basic equations of the proposed method and Section 3 explains all its steps. Finally, experiments are presented and discussed in Section 4.

2 Geometric background and basic equations

The reconstruction problem consists of 2 types of parameters, the 3D coordinates of the scene's points and the projection matrices. Since we are using uncalibrated images, solving the reconstruction problem means calculating all its parameters. Throughout this paper, we only suppose that we are given at least 2 images with point correspondences and a nonaccurate estimate of the intrinsic parameters' values.

Because the geometry of the camera is crucial to our method, this section starts with a quick review of the camera model we used, that is the pinhole model. Then, the basic equations of the reconstruction are derived.

2.1 The camera geometry

The pinhole model (see Figure 1) is a good approximation of a real camera and is by far the most used for modeling cameras. In this model, a pixel p on the image plane is the pure perspective projection of the point P of the scene. Using homogeneous coordinates, the pure perspective projection of the scene point $P = (X, Y, Z, T)$ on the image point $p = (x, y, t)$ can be represented by the relation

$$\begin{pmatrix} x \\ y \\ t \end{pmatrix} = \lambda \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \quad (1)$$

where λ is a scale factor. Note that we can assume that $t = T = 1$ in the Euclidean case.

[1] The term calibration here, means estimating the internal parameters (called also intrinsic parameters) of the camera.

[2] The camera's intrinsic parameters must remain unchanged through the different images.

Because the scene coordinate system and the camera coordinate system are not the same, the scene points must undergo a rigid displacement (a rotation and a translation) before the perspective projection. Furthermore, the image coordinate system uses pixels as units and uses the top-left corner of the image plane as origin. Hence, after projection, the points must undergo a scaling and a 2D translation. The above relation becomes

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

where α_u and α_v are the scale factors along the image x-axis and y-axis respectively and, (u_0, v_0) are the image coordinates of the intersection of the optical axis with the image plane (see Figure 1). These 4 parameters are called the intrinsic parameters.

A compact version of the above is given by

$$p = \lambda AIDP = \lambda MP \qquad (M = AID) \quad (3)$$

where M is called the projection matrix, A the intrinsic parameter matrix[3], I the pure perspective projection and D the displacement matrix.

A simple count of the parameters gives 10 parameters for the projection matrix M, that is, 6 extrinsic parameters and 4 intrinsic parameters.

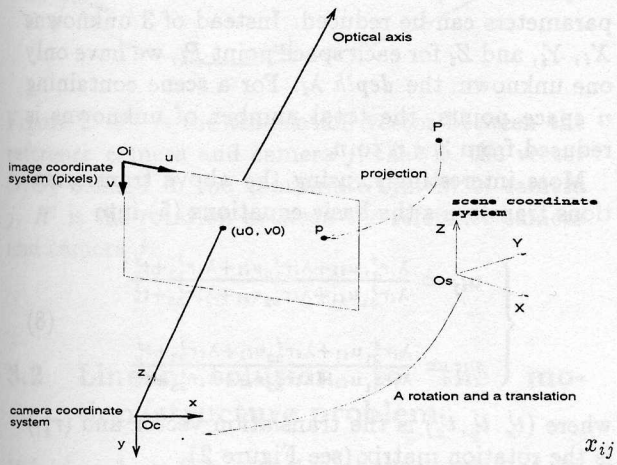

Figure 1: Camera geometry under the pinhole model

2.2 The basic equations

The 3D reconstruction is only possible when at least 2 different images of the same rigid scene are available. For simplicity, we assume in the following that we are given 2 images of a still scene obtained with a single camera that has undergone a displacement. That is, first an image of the scene is taken, the camera moves, and then the second image is taken by this same camera.

In the case of a camera with nonchanging intrinsic parameters, the projection matrix M_j of the j^{th} image is given by

$$M_j = AID_j$$

where

$$D_j = \begin{pmatrix} r^j_{11} & r^j_{12} & r^j_{13} & t^j_x \\ r^j_{21} & r^j_{22} & r^j_{23} & t^j_y \\ r^j_{31} & r^j_{32} & r^j_{33} & t^j_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Because we have the freedom to select a coordinate system for the scene, the first camera can be chosen as a reference camera. Thus, the coordinate system of the first camera is also the one of the scene. This choice yields a simplified projection matrix M_1 of the first image. In particular, M_1 has only 4 parameters:

$$M_1 = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Using the pinhole model, a space point $P_i = (X_i, Y_i, Y_i, 1)$ is projected in the j^{th} image on the point $p_{ij} = (x_{ij}, y_{ij}, 1)$ where

$$p_{ij} = \lambda M_j P_i \quad (4)$$

The above relation translates into the following two equations

$$x_{ij} = \frac{(\alpha_u r^j_{11} + u_0 r^j_{31})X_i + (\alpha_u r^j_{12} + u_0 r^j_{32})Y_i + (\alpha_u r^j_{13} + u_0 r^j_{33})Z_i + \alpha_u t_x + u_0 t_z}{r^j_{31}X_i + r^j_{32}Y_i + r^j_{33}Z_i + t_z}$$

$$y_{ij} = \frac{(\alpha_v r^j_{21} + v_0 r^j_{31})X_i + (\alpha_v r^j_{22} + v_0 r^j_{32})Y_i + (\alpha_v r^j_{23} + v_0 r^j_{33})Z_i + \alpha_v t_y + v_0 t_z}{r^j_{31}X_i + r^j_{32}Y_i + r^j_{33}Z_i + t_z} \quad (5)$$

We call the above equations the basic equations of the reconstruction problem. In particular, we can see that each image point provides 2 equations. Therefore, given a scene of n points and given m images of that scene provides us with $2 \times n \times m$ equations. A count of the parameters on the other hand yields $3 \times n$ for the 3D coordinates of the scene's points, plus $6 \times (m-1)$

[3] We made the common assumption that the x-axis and y-axis of the image plane are perpendicular.

for the displacements (the first camera is used as a reference camera making D_1 an identity matrix), and 4 intrinsic parameters. In theory, this problem has a solution when the number of equations is greater than the number of parameters, that is when $2 \times n \times m \geq 3 \times n + 6 \times (m-1)$. When the intrinsic parameters are not known, at least 3 images are required to solve for all the unknowns[15, 13].

The above equations are highly nonlinear making it nearly impossible to solve the reconstruction problem using standard optimization methods. Some recursive algorithms to solve for the above equations (or a simplified version of these equations) were proposed, see for instance [1] and [20]. However, these methods require a large number of images, typically several dozens. To our knowledge, no method has been reported in the literature that solves these equations in the general case using a minimum number of images.

The next section explains how these equations can be simplified and how they can be used for solving the reconstruction problem.

3 Solving the reconstruction problem

For any CCD camera, the intrinsic parameters are not completely unknown. Usually the camera's manufacturer provides their values but they are inaccurate. In addition, even if the manufacturer's values are not available, an estimate of these values is often available from previous experiments. Because these values are just an estimate of the exact values, using them usually yields unstable calculations. In particular, it is well known that noise in image points makes the calculation of the essential matrix very difficult[4]. If in addition inaccurate values for the intrinsic parameters are used, the calculated essential matrix will be practically useless. Therefore, we have avoided recovering the motion/structure through the calculation of the essential matrix. Instead, we have used another parameterization of the problem where both motion and structure are calculated in the same time using a minimum number of unknowns.

3.1 Simplifying the basic equations

We propose here another parameterization of the reconstruction problem that simplifies the above basic equations. As a consequence, even with inaccurate values for the intrinsic parameters, the calculations are more stable and the obtained reconstruction and camera motion are very close to the exact ones.

Let p be an image point given by its pixel coordinates (x, y). The normalized coordinates (u, v) of p, that is the 2D coordinates of p given in the camera's coordinate system are given by

$$u = \frac{x-u_0}{\alpha_u} \quad \text{and} \quad v = \frac{y-v_0}{\alpha_v} \tag{6}$$

where (u_0, v_0) are the pixel coordinates of the center of the image (intersection of the optical center with the image plane) and, α_u and α_v are the scale factors along the x-axis and y-axis.

Because we have used the first camera as a reference camera, the vector $O_1\vec{p}_{i1}$ and the vector $O_1\vec{P}_i$ are collinear for any space point P_i (see Figure 2). Therefore, the 3D coordinates (X_i, Y_i, Z_i) of a space point P_i can be written as a function of the normalized coordinates (u_{i1}, v_{i1}) of the image point p_{i1}, which is the projection in the first image (reference camera) of the point P_i. This translates into the relation

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \lambda_i \begin{pmatrix} u_{i1} \\ v_{i1} \\ 1 \end{pmatrix} \tag{7}$$

where λ_i is a scale factor representing the *depth* of P_i.

Using equation (6) and an estimate of the intrinsic parameters, the total number of the structure's parameters can be reduced. Instead of 3 unknowns X_i, Y_i, and Z_i for each space point P_i, we have only one unknown, the *depth* λ_i. For a scene containing n space points, the total number of unknowns is reduced from $3 \times n$ to n.

More interestingly, using the above transformations transforms the basic equations (5) into

$$\begin{cases} u_{ij} = \frac{\lambda_i r^j_{11} u_{i1} + \lambda_i r^j_{12} v_{i1} + \lambda_i r^j_{13} + t^j_x}{\lambda_i r^j_{31} u_{i1} + \lambda_i r^j_{32} v_{i1} + \lambda_i r^j_{33} + t^j_z} \\ v_{ij} = \frac{\lambda_i r^j_{21} u_{i1} + \lambda_i r^j_{22} v_{i1} + \lambda_i r^j_{23} + t^j_y}{\lambda_i r^j_{31} u_{i1} + \lambda_i r^j_{32} v_{i1} + \lambda_i r^j_{33} + t^j_z} \end{cases} \tag{8}$$

where (t^j_x, t^j_y, t^j_z) is the translation vector and (r^j_{kl}) is the rotation matrix (see Figure 2).

Because λ_i is the depth of the point P_i it cannot be zero. Hence, by dividing the numerators and denominators of the above equations by λ_i we obtain

$$\begin{cases} u_{ij} = \frac{r^j_{11} u_{i1} + r^j_{12} v_{i1} + r^j_{13} + \beta_i t^j_x}{r^j_{31} u_{i1} + r^j_{32} v_{i1} + r^j_{33} + \beta_i t^j_z} \\ v_{ij} = \frac{r^j_{21} u_{i1} + r^j_{22} v_{i1} + r^j_{23} + \beta_i t^j_y}{r^j_{31} u_{i1} + r^j_{32} v_{i1} + r^j_{33} + \beta_i t^j_z} \end{cases} \tag{9}$$

where $\frac{1}{\lambda_i} = \beta_i$.

It can be shown easily that the denominators of the above equations cannot be zero except for the

points of the plane defined by $Z = 0$. However, these points cannot be observed as they are not located in front of the camera. Therefore, we can transform equations (9) into

$$\begin{cases} u_{ij}(r^j_{31}u_{i1} + r^j_{32}v_{i1} + r^j_{33} + \beta_i t^j_z) = r^j_{11}u_{i1} + r^j_{12}v_{i1} + r^j_{13} + \beta_i t^j_x \\ v_{ij}(r^j_{31}u_{i1} + r^j_{32}v_{i1} + r^j_{33} + \beta_i t^j_z) = r^j_{21}u_{i1} + r^j_{22}v_{i1} + r^j_{23} + \beta_i t^j_y \end{cases} \tag{10}$$

The above equations are a lot simpler than equations (5). However, they are still nonlinear in three terms; namely, $\beta_i t^j_x$, $\beta_i t^j_y$ and $\beta_i t^j_z$. This nonlinearity is easier to handle, as shown in the next paragraph.

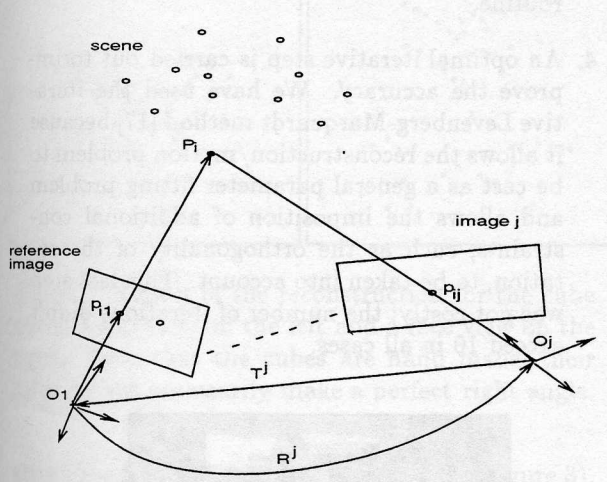

Figure 2: T^j is the translation vector between the reference camera and camera j; that is, the vector $\vec{O_j O_1}$ defined in the coordinate system of camera j. R^j is the rotation between the reference camera and camera j.

3.2 Linear solution to the motion/structure problem

It is clear from Figure 2 that the projection of the point O_1 on the jth image is the epipole of the latter with respect to the first image. Linear methods exist for calculating the epipoles from point correspondences (at least 8 correspondences) in the images. The most recent linear methods (see for instance [6] and [2]) yield excellent results that are comparable to the results obtained by nonlinear methods.

Without loss of generality, let's consider here the case of two images where the first one is used as a reference image (a reference camera).

Calculating the translation T^2

Let $e = (e_x, e_y, e_t)$ be the homogeneous coordinates of the epipole in the second image (in this work we have used the method developed by ourselves [2] to calculate e). Because the point O_1 is the origin of the reference coordinate system (the coordinate system of the first camera and of the scene), its homogeneous coordinates are $(0, 0, 0, 1)$. O_1 is projected on the point e in the second image by the relation

$$\begin{pmatrix} e_x \\ e_y \\ e_t \end{pmatrix} = \lambda M_2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{11}$$

where λ is a scale factor and M_2 is the projection matrix of the second image (see relation (2)).

After developing the above, we obtain

$$\begin{pmatrix} e_x \\ e_y \\ e_t \end{pmatrix} = \lambda A \begin{pmatrix} t^2_x \\ t^2_y \\ t^2_z \end{pmatrix} \tag{12}$$

where A is the 3×3 matrix of the intrinsic parameters and (t^2_x, t^2_y, t^2_z) are the coordinates of the translation vector T^2 (don't read T^2 as T squared, but as T for image 2). The above relation yields

$$\begin{pmatrix} e_x \\ e_y \\ e_t \end{pmatrix} = \lambda \begin{pmatrix} \alpha_u t^2_x + u_0 t^2_z \\ \alpha_v t^2_y + v_0 t^2_z \\ t^2_z \end{pmatrix} \tag{13}$$

In order to solve for the unknowns t^2_x, t^2_y and t^2_z, let's consider 2 cases. The first case, when $e_t = 0$, implies that $t^2_z = 0$. And the second case, when $e_t \neq 0$, implies that $t^2_z \neq 0$. Note that e is supposed to be known (calculated from at least 8 point correspondences).

- Case 1: $t^2_z = 0$. Since relation (13) is an equality up to a scale factor, the only way to get rid of the scale factor is by using ratios. In particular, from (13) we can write

  $$\frac{\alpha_u}{\alpha_v} e_y t^2_x - e_x t^2_y = 0$$

  Note that even though α_u and α_v are approximately known, the value of the ratio $\frac{\alpha_u}{\alpha_v}$ is very stable (even with a zoom) and is usually accurately known.

  The above equation is not enough to solve for the unknowns t^2_x and t^2_y. However, we know

that it is impossible to determine the norm of the translation T^2 because of the distance-size ambiguity. In other words, only 2 parameters of T^2 can be determined. Hence, we can suppose that $t_x^2 = 1$ (if e_x is nonzero) or $t_y^2 = 1$ (if e_y is nonzero).

- Case 2: $t_z^2 \neq 0$. Because of the distance-size ambiguity, we can take $t_z^2 = 1$. Relation (13) yields 2 linear equations in the unknowns t_x^2 and t_y^2.

Therefore, the translation T^2 can be calculated separately. Equations (10) become linear making it possible to calculate all the parameters of the motion/structure problem.

Note: The translation vector T^2 can be calculated only up to a sign factor, that is, (t_x^2, t_y^2, t_z^2) and $(-t_x^2, -t_y^2, -t_z^2)$ can both be solutions. This duality corresponds to the existence of two compatible solutions for the motion/structure problem [14]. These 2 solutions correspond to two different reconstructions; in one case the reconstructed points are located in front of the camera (all β_i are positive) and in the other case, the reconstructed points are located behind the camera (all β_i are negative). The only way to solve this problem is by calculating both solutions and discarding the wrong one ($\beta_i \leq 0$). This is a necessary but easy task since all steps of our algorithm are linear.

3.3 A nonlinear step

Although we can be satisfied with our linear algorithm, adding a nonlinear optimal step improves the results. We see at least two reasons for using a nonlinear step; the first one is that it is not costly since we have a very good estimation of all parameters, and the second reason is that it allows the constraints on the rotation (the rotation matrix is orthogonal and has only 3 independent parameters) to be added.

4 Experiments

Two different experiments have been carried out. The first one aims to validate our approach and to show that qualitative reconstructions can be obtained. The second one aims to show the accuracy we obtain on the reconstruction and on the motion. Although we have used only 2 images for each experiment, the algorithm can be used with any number of images without major changes.

4.1 Summary of our algorithm

Given at least 8 point correspondences in 2 images and an estimation of the intrinsic parameters, the reconstruction problem is defined by equations (10).

The reconstruction method presented in this paper can be summarized by the following sequential steps.

1. The epipole in the second image is calculated using the method described in [2].
2. The translation vector is then calculated.
3. The rotation matrix $(r_{11}, ..., r_{33})$ and the structure parameters β_i are calculated using an SVD routine.
4. An optimal iterative step is carried out to improve the accuracy. We have used the iterative Levenberg-Marquardt method [17] because it allows the reconstruction/motion problem to be cast as a general parameter fitting problem and allows the imposition of additional constraints, such as the orthogonality of the rotation, to be taken into account. This last step was not costly; the number of iterations didn't exceed 10 in all cases.

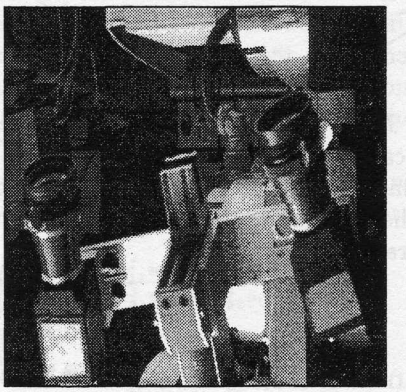

Figure 3: The stereo head: the 2 cameras at a distance of 35cm from each other.

4.2 First experiment: testing the quality

The scene consists of 2 cubes put one upon the other (see Figure 4). Polygonal motifs were stuck on the cubes to provide for interest points (corners of those polygons were our interest points). A total of 49 interest points were extracted and matched in the two images. The two images were obtained with the

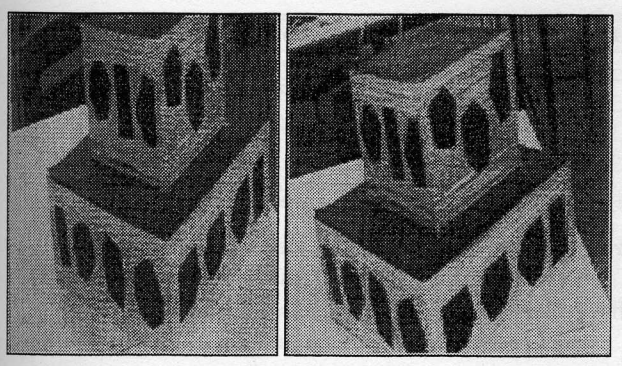

Figure 4: The stereo pair of images (cubes).

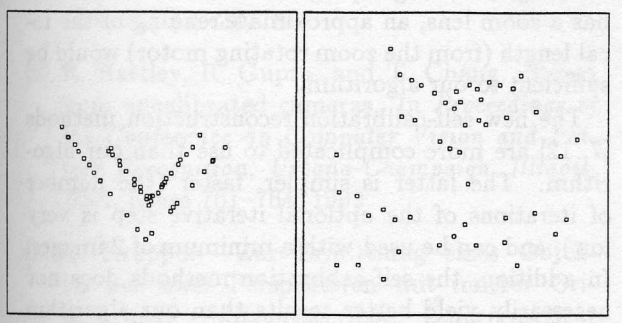

Figure 5: Result of the reconstruction for the cube scene; a top view on the left and a face view on the right. Note that the cubes are hand made, their sides do not necessarily make a perfect right angle.

stereo head mounted on a robot arm[4](see Figure 3). We used the same intrinsic parameters for the two cameras. The values used are given by

$$A = \begin{pmatrix} -1500 & 0 & 256 \\ 0 & 1000 & 256 \\ 0 & 0 & 1 \end{pmatrix} \qquad (14)$$

The above values are not accurate. They were chosen because of our experience with the cameras.

47 matched interest points were the input of our algorithm. The output was the reconstructed scene shown on Figure 5. As one can see from the top view, the quality of the reconstruction is excellent. In particular, the coplanarity of the points is visible.

4.3 Second experiment: testing the accuracy

We used in this experiment a known calibration pattern. The interest points are the corners of the black squares (see Figure 6). The calibration pattern is placed on a micrometric table which can undergo

[4]The images were taken at the INRIA lab of Grenoble, France.

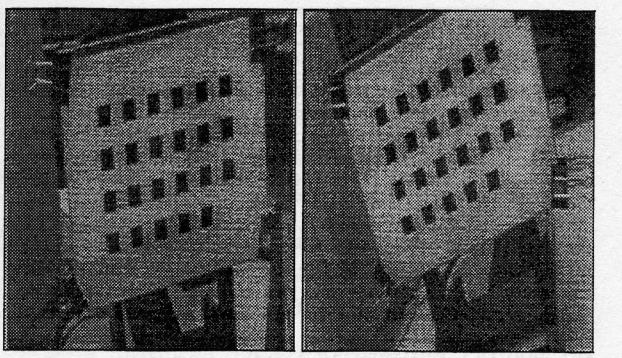

Figure 6: The calibration patter (stereo images)

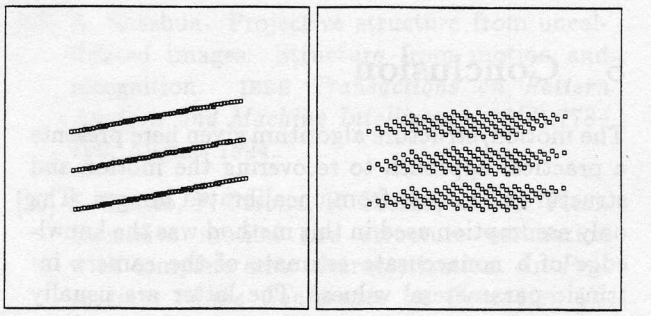

Figure 7: Reconstruction of the calibration pattern.

pure translations. We can hence obtain a nonplanar scene by using several images of the calibration pattern at different locations. In our case, one image consists of 3 planes.

In this experiment 276 matched points were used as an input to our algorithm. Using the same intrinsic parameters with the values given in (14), the obtained reconstruction is shown in Figure 7.

In order to test the accuracy of our algorithm, we calibrated our cameras and compared the motion we obtained with the one obtained through the calibration. The motion here corresponds to the orientation of one camera with respect to the other. Note that in this case it was possible to calibrate the cameras because we knew exactly the 3D coordinates of the calibration pattern's points. The results we obtained are summarized in Table 1. Using the values obtained through the calibration ("calibration" entry in Table 1) as reference values, Table 1 shows that despite a 10% error in the intrinsic parameters (values assigned to those parameters) the motion we computed using our algorithm remains of an acceptable quality. Note that we didn't claim that our algorithm will do as good as a calibration-based algorithm. However, even with poorly accurate intrinsic parameters, the algorithm is able to recover a good quality motion/structure. The latter can

	Rotation's angles			*Translation*		*intrinsic parameters*			
	α	β	γ	t_x/t_z	t_y/t_z	α_u	α_v	u_0	v_0
Calibration	-23.56	4.24	-9.83	-0.995	-3.135	-1466.22	997.70	234.38	242.03
Test1	-23.57	6.01	-8.88	-0.884	-2.908	-1500	1000	256	256
Test2	-25.75	7.58	-7.37	-0.820	-2.881	-1600	1100	256	256
Test3	-24.03	2.37	-10.43	-0.846	-2.824	-1500	1000	280	280

Table 1: The "calibration" entry shows the values calculated using the classical calibration method. The entries *Test1*, *Test2* and *Test3* show the motion (rotation and translation) we obtained when using different values for the intrinsic parameters.

be used for applications where high accuracy is not required.

5 Conclusion

The motion/structure algorithm given here presents a practical approach to recovering the motion and structure of a scene from uncalibrated images. The only assumption used in this method was the knowledge of a nonaccurate estimate of the camera intrinsic parameters' values. The latter are usually known either from the manufacturer's data or from previous experiments.

Using a different parameterization for the motion/structure problem has resulted into a significant reduction of the total number of parameters. In addition, the basic equations of the problem became simpler. This simplicity together with the simultaneous calculations of the motion and the structure made the computations very stable. As a consequence, we obtained good quality results even with poorly accurate intrinsic parameters' values.

Classical methods based on the essential matrix suppose the knowledge of the exact values of the extrinsic parameters. The latter are usually obtained through the explicit calibration of the camera. Those methods would not perform well with poorly accurate intrinsic parameters' values. In particular, the calculation of the essential matrix is very sensitive to pixel errors and its decomposition into translation and rotation amplifies this sensitivity. In addition, as the motion and the structure are not simultaneously calculated, the sensitivity to all errors (pixel errors and intrinsic parameters' errors) is even more amplified making the results useless.

Naturally, the algorithm in this paper cannot hope to give results as accurate as those obtained with calibrated cameras where, the calibration process uses calibration grids. Nevertheless, the algorithm is adequate for applications where high accuracy in the motion/structure is not required. For instance, the algorithm would be of greatest use where euclidean scene reconstruction is used for purposes of navigation, or grasping. Even when the camera has a zoom lens, an approximate reading of the focal length (from the zoom rotating motor) would be sufficient to our algorithm.

The new self-calibration reconstruction methods [7, 12] are more complicated to use than our algorithm. The latter is simpler, faster (the number of iterations of the optional iterative step is very low), and can be used with a minimum of 2 images. In addition, the self-calibration methods does not necessarily yield better results than our algorithm (a comparison study will tell). Thus, practically speaking our algorithm might be preferred over a reconstruction method based on self-calibration.

References

[1] A. Azarbayejani, B. Horowitz, and A. Pentland. Recursive estimation of structure and motion using relative orientation constraints. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA*, pages 294–299, June 1993.

[2] B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1030–1036, June 1995.

[3] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 563–578. Springer-Verlag, May 1992.

[4] O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint.* Artificial intelligence. M.I.T. Press, Cambridge, MA, 1993.

[5] O.D. Faugeras, Q.T. Luong, and S.J. Maybank. Camera self-calibration: Theory and experiments. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 321–334. Springer-Verlag, May 1992.

[6] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1064–1070, June 1995.

[7] R. Hartley. Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1):5–23, 1997.

[8] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Urbana-Champaign, Illinois, USA*, pages 761–764, 1992.

[9] E. Kruppa. Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. *Sitzungsberichte Österreichische Akademie der Wissenschaften, Mathematisch-naturwissenschaftliche Klasse, Abteilung II a*, 122:1939–1948, 1913.

[10] C.H. Lee and T. Huang. Finding point correspondences and determining motion of a rigid object from two weak perspective views. *Computer Vision, Graphics and Image Processing*, 52:309–327, 1990.

[11] H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. In *Nature*, volume 293, pages 133–135. XX, September 1981.

[12] Q.T. Luong. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289, 1997.

[13] Q.T. Luong and O.D. Faugeras. Self-calibration of a camera using multiple images. In *Proceedings of the 11th International Conference on Pattern Recognition, The Hague, Netherland*, pages 9–12, 1992.

[14] S. Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, 1993.

[15] 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.

[16] T. Moons, L. van Gool, M. van Diest, and E. Pauwels. Affine reconstruction from perspective image pairs. In *Proceeding of the* DARPA–ESPRIT *workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pages 249–266, October 1993.

[17] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[18] C.A. Rothwell. *Object Recognition Through Invariant Indexing*. Oxford Science Publications, 1995.

[19] A. Shashua. Projective structure from uncalibrated images: Structure from motion and recognition. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(8):778–790, August 1994.

[20] S. Soatto, P. Perona, R. Frezza, and G. Picci. Recursive motion and structure estimation with complete error characterization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA*, pages 428–433, June 1993.

[21] C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. In *Proceedings of the* IEEE *Workshop on Visual Motion, Princeton, New Jersey*, pages 21–28, Los Alamitos, California, USA, October 1991. IEEE Computer Society Press.

[22] L. van Gool, T. Moons, M. Proesmans, and M. van Diest. Affine reconstruction from perspective image pairs obtained by a translating camera. In *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel*, volume I, pages 290–294. Proceedings of IAPR Workshop on Computer Vision, IEEE Computer Society Press, October 1994.

[23] D. Weinshall. Model-based invariants for 3D vision. *International Journal of Computer Vision*, 10(1):27–42, 1993.