Three-dimensional structure calculation : achieving accuracy without calibration

B. Boufama and A. Habed
SCHOOL OF COMPUTER SCIENCE
University of Windsor
Windsor, Ontario
Canada N9B 3P4

Abstract

This paper addresses the problem of computing the camera motion and the three-dimensional structure of a scene using two uncalibrated images as inputs. The camera motion is calculated by estimating the essential matrix and using approximate values, easily available, for the intrinsic parameters. The classical eight-point algorithm to calculate the essential matrix is known to be very sensitive to pixel-noise even when the intrinsic parameters are perfectly known. This paper shows that by using the normalized eight-point algorithm, aimed at calculating the fundamental matrix, the pixel-noise sensitivity is reduced significantly. More importantly, we show that the intrinsic parameters do not have to be accurately known in order to get very good quality reconstruction. In particular, we have investigated and compared the effect of errors on the intrinsic parameters together with pixel-noise on the calculated motion/structure, when using the straightforward eight-point algorithm and its normalized version respectively.

keywords: three-dimensional reconstruction, camera motion, essential matrix.

1 Introduction

Our surrounding environment is three-dimensional (3D) by nature. This 3D structure is very important and is often a prerequisite for other subsequent activities such as navigation, 3D modeling, surgery, etc. However, the most common way to capture this environment is through two-dimensional (2D) images. In computer vision, three-dimensional reconstruction of a scene is the process of recovering/computing the 3D coordinates (or depth) of the scene's points using 2D images as inputs.

Although the projective 3D structure of a scene can be recovered from point correspondences only [3] [13] [7], such a structure lacks metric information making its use limited. The Euclidean structure remains the most useful one 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 nonrealistic and it is not always possible. 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 [11] [15] [16]. These methods have their limitations. In particular, they assume that the size of the scene is small compared to the scene-camera distance.

Another direction for solving the Euclidean reconstruction has focused on the self-calibration of the camera, where matched points in the images are used to estimate the intrinsic parameters [5]. Once the camera is calibrated, the problem of calculating the motion of the camera/scene and the Euclidean reconstruction of the observed scene becomes easier [12] and can be calculated by triangulation or by a least-squares algorithm.. However, the proposed self-calibration methods are nonlinear and do not yield accurate values for the intrinsic parameters.

This paper shows that the eight-point algorithm can be modified into a more stable and robust algorithm. As it was done for the calculation of the fundamental matrix[6], we applied the same type normalization and used the eight-point to calculate the essential matrix. In this method, we do not extract the essential matrix from the fundamental matrix which is not needed here. Instead, we use approximate values for the intrinsic paremeters, change pixel coordinates to the unit sphere coordinates, apply the normalization on these coordinates and, finally calculate the essential matrix using an SVD routine. Our main goal is to show that the intrinsic parameters need not to be precisely known in order to get an accurate three-dimensional Euclidean reconstruction.

As expected, this normalized eight-point algorithm has proven to be less sensitive to pixel noise. However, less expected, this algorithm has also proven to be less affected by

errors on the intrinsic parameters. In particular, the relative reconstruction can be achieved with very good accuracy even when the intrinsic parameters are no better than a good guess.

2 Longuet-Higgins' constraint : eight-point algorithm

The first method to calculate the camera relative motion was proposed by Longuet-Higgins [12]. In this method, the essential matrix is calculated from pixel correspondences using a simple and straightforward linear algorithm, when the intrinsic parameters are known. Newer methods based on the same idea and taking into account noisy data were developed later(see for instance [17]).

Because Longuet-Higgins' algorithms [12] carries the basic ideas used by most 3D reconstruction solutions[9], we present in the following paragraph a review of this algorithm. In addition, an improved version for Longuet-Higgins' algorithm is presented in next paragraph.

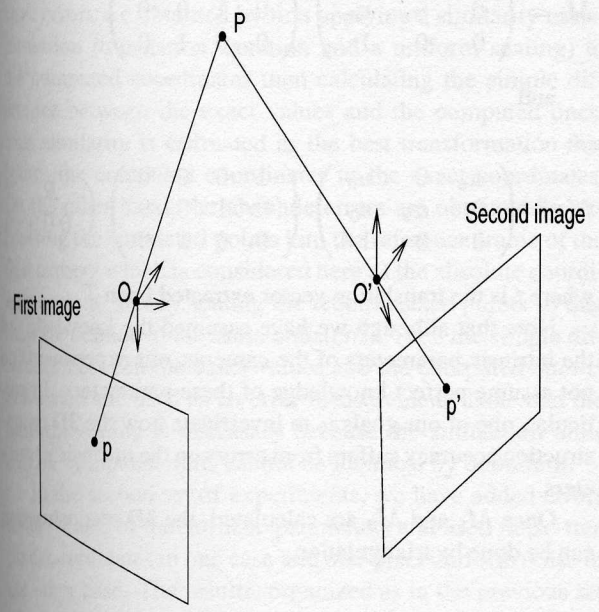


Figure 1: P, p, O, p' and O' are all coplanar

Consider a space point P and its projections, p and p', on the first and second image respectively (see Figure 1). It is clear from Figure 1 that the three vectors $\vec{Op}$, $\vec{Op'}$, and $\vec{O'O}$ are coplanar. Therefore, each couple of matched points (p,p'), given by their normalized coordinates, in 2 images yields the following relation :

$$\vec{O'p'} \cdot (\vec{O'O} \wedge \vec{Op}) = 0 \tag{1}$$

where $\cdot$ denotes the scalar product and $\wedge$ denotes the cross product.

However, the above relation is only true when all vectors are defined in the same reference frame. Because we are free to choose any reference frame, let's take the one attached to the second camera as our reference frame. This choice transforms the above relation into :

$$\vec{O'p'} \cdot (\vec{O'O} \wedge R\vec{Op}) = 0 \tag{2}$$

where R is the 3×3 rotation matrix from the reference frame of the second camera to the one of the first camera.

Let's denote $\vec{O'p'}$, $\vec{O'O}$ and $\vec{Op}$ by p'^T, t and p respectively, equation (2) becomes :

$$p'^T \cdot ([t]_\times Rp) = p'^T TRp = 0 \tag{3}$$

where $t = (t_x, t_y, t_z)$, the translation vector, represents the coordinates of O in the reference frame of the second camera and T is the 3×3 antisymmetric matrix given by

$$T = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$$

the above relationship is usually written as

$$p'^T Ep = 0 \tag{4}$$

where E is a 3×3 matrix known as the *essential matrix*.

E has exactly five degrees of freedom since the norm of t cannot be retrieved due to the distance/scale ambiguity. It describes the Euclidean motion (rotation and translation) between two cameras. Note that the *fundamental matrix* is a generalization to the uncalibrated case(pixel coordinates) of the *essential matrix*.

The above relation is known as Longuet-Higgins' constraint. Because E is defined up to a scale factor, it can be computed by solving a set of linear equations given eight or more point correspondences. This is known as the Eight-Point Algorithm. The calculation of E using the linear equations given in (4) was very sensitive to noise yielding inaccurate motion and structure. It was believed that since E has only five degrees of freedom, nonlinear constraints on E must be enforced [4]. Hence, several nonlinear techniques were proposed(see for instance [1] and [14]). However, such techniques come with two drawbacks, their convergence to a global minimum is not ensured and their complexity is too high.

3 Improved Eight-Point Algorithm

Two recent methods for calculating the *fundamental matrix* F [6][2] have shown that with minor transformations on the image coordinates, the calculated F using the Eight-Point Algorithm can be as good as the best nonlinear algorithm. These two linear algorithms can be viewed as an improved

Eight-point algorithm. Although they were developed for calculating the *fundamental matrix*, the same idea can be used and applied for the calculation of the *essential matrix*. Basically, instead of using the pixel coordinates of a point given by the vector $p = (x, y, 1)$, we use the coordinates given by $A^{-1}p$, where A is a 3×3 matrix containing the intrinsic parameters.

3.1 Normalized image coordinates method

Hartley [6] has shown that the high sensitivity to image noise of the eight-point algorithm is due to the poor numerical condition of the linear equations. He proposed to perform a simple normalization of the image coordinates (input data) prior to running the classical eight-point algorithm. One way to make this normalization of the data is summarized below:

1. The coordinates in the image are translated so that their centroid is at the origin.
2. Then, these coordinates are scaled, so that the average distance of points from the origin is equal to $\sqrt{2}$.

Such a transformation is applied to each of the two images independently.

Thanks to this normalization, the calculation process was more stable and yielded more accurate results as shown in Section 5.

3.2 Virtual parallax method

In [2] a new method, based on virtual parallax, was proposed to calculate the fundamental matrix F given by $[e']_\times H$, where $e' = (e_x, e_y, e_z)$ is the epipole in the second image and H is a plane homography. In this method the fundamental matrix is computed by calculating H and e', reducing therefore the number of parameters to be estimated. A quick summary of this method is given below.

Without loss of generality, a change of projective coordinates is performed in each image such as:

$$\begin{array}{ll} p_1 = (0,0,1)^T & p'_1 = (0,0,1)^T \\ p_2 = (1,0,0)^T & p'_2 = (1,0,0)^T \\ p_3 = (0,1,0)^T & p'_3 = (0,1,0)^T \\ p_0 = (1,1,1)^T & p'_0 = (1,1,1)^T \end{array}$$

In each image, these 4 points must form a projective basis, i.e., no three of them are collinear. The first three points define a plane in space. Under such choice of coordinate systems, the homography H such that $p'_i \simeq Hp_i (i = 1, 2, 3)$ is diagonal, that is, $H = diag(\alpha, \beta, \gamma)$, and depends only on two parameters.

For each additional match (p, p'), given by their homogeneous coordinate vectors $(x, y, t)^T$ and $(x', y', t')^T$ respectively, we have $(e' \times p') \cdot Hp = 0$ (Hp is on the line $< e'p' >$). That is

$$(e'_t y' - t' e'_y)\alpha x + (t' e'_x - e'_t x')\beta y + (e'_y x' - y' e'_x)t = 0 \quad (5)$$

This is the basic epipolar equation based on the virtual parallax. Using a few simple substitutions and adding one extra variable, the above equation can be linearized.

4 Three-dimensional reconstruction

Calculating E is not enough for the 3D reconstruction process. First, R and T must be extracted from the calculated E. This is done by factoring E into the product TR of a skew-symmetric matrix and a rotation matrix [8]. Then, given R and T, the two projection matrices, M_1 and M_2, associated with the first and the second camera, respectively, can to be computed. Without loss of generality, the scene coordinate system can be assumed to be the coordinate system attached to the first camera. Therefore, we have

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

and

$$M_2 = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & t \\ 0\,0\,0 & 1 \end{pmatrix}$$

where t is the translation vector extracted from T.

Note that although we have assumed the knowledge of the intrinsic parameters of the cameras, our experiments do not assume perfect knowledge of these parameters. In particular, one of our goals is to investigate how the 3D reconstruction accuracy suffers from errors on the intrinsic parameters.

Once M_1 and M_2 are calculated, the 3D reconstruction can be done by triangulation.

5 Experiments

Both the Eight-Point Algorithm and its improved version were implemented using an SVD solution. The latter is the method of choice for solving most linear least-squares problems. It has the advantage of being numerically stable and taking the presence of noise into account. Once the essential matrix E was calculated, the rotation matrix and the translation vector were extracted using a method based on [8].

We have carried out experiments on both simulated and real data. Two issues were investigated, the effect of pixel noise and the effect of the intrinsic parameters noise on the 3D reconstruction.

5.1 Tests on simulated data

A scene of 60 points covering a volume of $30cm \times 30cm \times 20cm$ was simulated. The virtual camera was located at 100cm off the scene and was assigned the following intrinsic parameters' values:

$$\begin{pmatrix} -1000 & 0 & 256 \\ 0 & -1000 & 256 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Two images of this scene were generated. The displacement between the two cameras consisted of a rotation and a translation given by $(5, 20, 0)$ and $(-25, 12, 12)$ respectively. The units for the angles and for the translation are degrees and centimeters respectively. These simulated data were used to carry out two sets of experiments. In the first set of experiments, we have used noisy pixel coordinates but the values of the intrinsic parameters were noise-free. Uniform noise, with magnitudes ranging from 0.2 pixel through 1.0 pixel, was added to the pixel coordinates. The results are summarized on Table 1. Each row of this table gives the mean errors for the corresponding noise magnitude. The relative errors are obtained by first applying a similarity transformation (translation, rotation and a uniform scaling) to the computed coordinates then calculating the simple difference between the exact values and the computed ones. This similarity is estimated as the best transformation that brings the computed coordinates to the exact coordinates. On the other hand, the absolute errors are obtained by expressing the simulated points into the reference frame of the first camera which is considered here as the absolute coordinate system, and by scaling the reconstructed points so that they are defined in the same units(cm). Then the simple difference between the exact values and the computed ones is calculated and used as an error measurement. Note that the uniform scaling is necessary because the simulation units and the computed units cannot be identical by definition.

In the second set of experiments, we have added errors to the values of the intrinsic parameters and used noise-free pixel coordinates in one case and one-pixel uniform noise in the other case. The results, organized as in the previous set of experiments, are summarized in Table 2 and Table 3.

5.2 Tests on real data

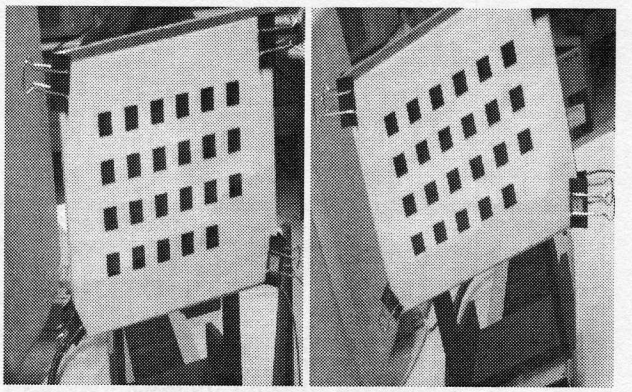
Because the tests with simulated data have proven the superiority of the normalized version of the eight-point algorithm, tests on real data were carried out using the improved eight-point algorithm only. Two scenes were used, the pattern scene and the house scene (see Figure 2).

1. Pattern scene: a known calibration pattern consisting of three parallel planes covering a volume of $18cm \times 12cm \times 6cm$ was placed at about $100cm$ off a stereo-rig. A total of 23 interest points were used here. These points were extracted from the images with a 0.1-pixel accuracy. Because the 3D coordinates of these points are known, the accuracy of the reconstruction can be reliably estimated.

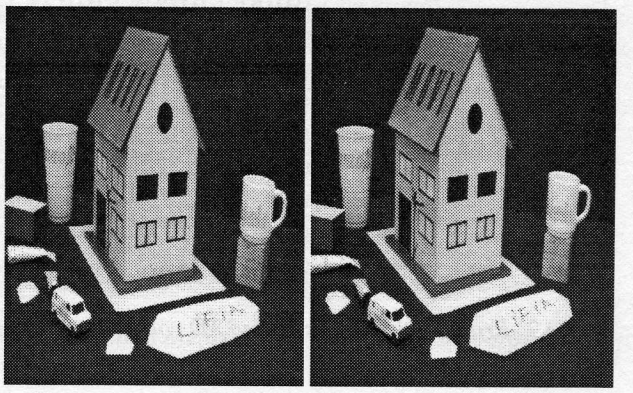
2. The house scene: This is a $40cm \times 40cm \times 25cm$ scene that was located at about 1 meter off the camera. A total of 38 points matched in two images were used. Unlike the pattern scene, the pixel location precision was not less than one pixel since we were using integer values. Furthermore, the calibration parameters were not available. However, because the same cameras were used, we have used the same values.

Table 4 summarizes the results of the reconstruction process for both scenes. Each row in this table gives the reconstruction errors for a given set of values assigned to the intrinsic parameters. Note that although the two views were taken by two different cameras(the same model), we have used the same values for the intrinsic parameters.

The values of the intrinsic parameters in the first row are the one obtained from the calibration process for the left camera. The values in the other rows were assigned close to the calibration values for testing purposes.



The Pattern image pair



The House image pair

Figure 2: The two pairs of real images used in the experiments.

noise level	method used	*Relative errors*				*Absolute errors*			
		ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance (cm)	ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance (cm)
0.2	classic	0.0116	0.0101	0.0438	0.0464	0.0172	0.0324	0.1588	0.1628
	improved	0.0122	0.0102	0.0436	0.0464	0.0205	0.0383	0.1314	0.1431
0.4	classic	0.0302	0.0468	0.1055	0.1193	0.0499	0.1512	0.5472	0.5847
	improved	0.0271	0.0277	0.0896	0.0976	0.0345	0.0563	0.4633	0.4713
0.6	classic	0.0642	0.0735	0.1241	0.1579	0.1799	0.1763	1.2457	1.3000
	improved	0.0457	0.0426	0.1131	0.1292	0.0903	0.0839	0.6305	0.6625
0.8	classic	0.1040	0.1224	0.1665	0.2313	0.2572	0.2286	1.6284	1.6999
	improved	0.0574	0.0588	0.1445	0.1662	0.1110	0.1026	0.8535	0.8865
1.0	classic	0.1717	0.2122	0.2118	0.3455	0.4502	0.4139	2.8088	2.9468
	improved	0.0791	0.0888	0.2074	0.2391	0.1390	0.1280	1.0559	1.1020

Table 1: *Errors in the reconstructed 3D coordinates when noise is added to the image coordinates while the intrinsic parameters are error-free(known).*

Noise on intrinsic parameters (%)	Method used	*Relative errors*				*Absolute errors*			
		ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance (cm)	ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance (cm)
2.5	Classic	0.0109	0.0213	0.0127	0.0271	0.2229	0.2864	2.9013	2.9354
	Improved	0.0113	0.0205	0.0125	0.0266	0.0813	0.3196	0.2299	0.4264
5	Classic	0.0169	0.0328	0.0203	0.0421	0.6236	0.9075	5.7145	5.8302
	Improved	0.0161	0.0308	0.0178	0.0391	0.4325	0.7878	0.3721	1.0577
10	Classic	0.0195	0.0452	0.0244	0.0550	0.7942	1.5735	2.8228	3.3848
	Improved	0.0216	0.0379	0.0256	0.0506	0.7118	1.7413	0.4182	2.0088
15	Classic	0.0365	0.0598	0.0786	0.1053	4.7281	3.2765	5.6204	8.1362
	Improved	0.0578	0.0773	0.0568	0.1119	4.3370	3.0669	1.1041	5.5282
20	Classic	0.0364	0.0628	0.0366	0.0812	2.0376	1.4487	11.6194	11.8853
	Improved	0.0443	0.0731	0.0498	0.0989	2.6421	0.9241	0.8933	3.0112
25	Classic	0.3986	0.6738	0.7430	1.0793	3.8877	5.2214	23.4350	25.2685
	Improved	0.2188	0.3875	0.2660	0.5184	4.7979	1.5534	4.5196	7.3749

Table 2: *Errors in the reconstructed 3D coordinates when noise is added to the intrinsic parameters while pixel coordinates are noise-free.*

Noise on intrinsic parameters (%)	Method used	*Relative errors*				*Absolute errors*			
		ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance (cm)	ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance (cm)
2.5	Classic	0.1803	0.2210	0.2116	0.3552	0.4601	0.7070	2.9490	3.1880
	Improved	0.0841	0.0978	0.2075	0.2443	0.1331	0.2983	1.1638	1.2512
5	Classic	0.1796	0.2066	0.3547	0.4480	0.4800	0.7807	3.1520	3.3968
	Improved	0.0755	0.0843	0.2078	0.2366	0.3928	0.8729	1.0481	1.5591
10	Classic	0.2023	0.2517	0.4041	0.5173	1.7352	1.1181	3.4490	4.3561
	Improved	0.0940	0.1127	0.2099	0.2561	0.6245	1.7343	1.4202	2.5093
15	Classic	0.6395	0.6751	1.5573	1.8138	3.4568	2.3653	2.2050	5.0290
	Improved	0.0518	0.0729	0.2114	0.2296	4.4503	2.9712	0.8883	5.5246
20	Classic	1.1048	0.9798	1.5146	2.1153	4.1636	4.3479	12.2381	14.2944
	Improved	0.0860	0.1156	0.2097	0.2544	2.7252	0.9833	1.2484	3.2712
25	Classic	0.9525	0.7842	1.1885	1.7131	6.6859	5.1887	45.3754	46.6645
	Improved	0.2275	0.3811	0.2951	0.5330	4.6926	1.4828	4.7449	7.3805

Table 3: *Errors in the reconstructed 3D coordinates when noise is added to the intrinsic parameters and pixels were added a 1-pixel uniform noise.*

Intrinsic parameters used	Scene	*Relative errors*				*Absolute errors*			
		ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance	ΔX (cm)	ΔY (cm)	ΔZ (cm)	average distance
-1483 -1007	Pattern	0.0101	0.0022	0.0161	0.0191	0.4306	0.2160	0.1150	0.4985
242 239	House	0.0511	0.0543	0.0357	0.0827	2.9061	4.6781	0.6515	5.5892
-1500 -1000	Pattern	0.0112	0.0070	0.0168	0.0214	0.1882	1.4474	0.1448	1.4711
256 256	House	0.0485	0.0509	0.0346	0.0784	2.6405	4.2488	0.6042	5.0856
-1600 -1050	Pattern	0.0080	0.0062	0.0177	0.0204	0.8522	0.8276	0.1551	1.2214
276 246	House	0.0552	0.0569	0.0390	0.0883	2.2693	4.4840	0.6448	5.1286
-1600 -900	Pattern	0.0295	0.0278	0.0255	0.0478	1.5649	0.6806	0.7724	1.9673
236 240	House	0.0418	0.0454	0.0329	0.0699	2.9410	4.7221	0.6185	5.6488
-1400 -900	Pattern	0.0223	0.0190	0.0228	0.0371	0.7919	1.4057	0.6184	1.8342
236 260	House	0.0368	0.0407	0.0288	0.0619	3.0676	4.1603	0.5617	5.2331
-1700 -1100	Pattern	0.0141	0.0046	0.0126	0.0195	0.5662	2.6578	0.3098	2.7760
250 270	House	0.0579	0.0573	0.0404	0.0909	2.6165	3.8957	0.6245	4.8119

Table 4: *Errors in the reconstructed 3D coordinates when approximate values are assigned to the intrinsic parameters for real images, pattern and house scenes.*

5.3 Discussion

It was widely accepted that the simple Eight-point algorithm is too sensitive to noise and is numerically unstable. This is partly due to the fact that accuracy of the reconstructed points was measured in the reference frame of one camera(absolute reference frame). From our tests, most of the errors seem to consist of a uniform shift that is absorbed by a similarity transform. In particular, the depth, which is not a relative measurement, is poorly estimated when noise level gets high. This is in agreement with what is commonly known about the difficulty to accurately recover the depth information.

From Table 1, one can see that the relative errors of both algorithms are similar for noise levels below 0.8 pixels. However, with one-pixel error, the improved eight-point algorithm performed much better with an average distance of $0.23cm$ instead of $0.34cm$. On the other hand, the improvement is significant for absolute errors. In particular, the improved eight-point algorithm is far more accurate with respect to depth. For instance, with one-pixel error, the depth error went down to $1.05cm$ from $2.80cm$ for the classical eight-point algorithm.

Table 2 again confirms the higher accuracy of the improved eight-point algorithm over the classic one. Furthermore, it shows that errors on the intrinsic parameters generate mostly absolute errors that are absorbed by a similarity transform. In particular, even with a 20% errors added to the intrinsic parameters (this is large error), the relative reconstruction remains very accurate with an average error below one $0.1cm$.

Table 3, where noise is added to both pixel coordinates and intrinsic parameters, is a more realistic situation in practice. The results show clearly that a 1-pixel noise on the image coordinates generates errors on the relative 3D reconstruction that offset errors generated from noise on the intrinsic parameters. As it can be seen from Table 1 and Table 3, the improved eight-point algorithm average relative error remained around $0.24cm$ when, pixel noise level was 1-pixel and intrinsic parameters noise ranged from 0% to 20%. With a 25% noise on the intrinsic parameters, the relative average error increased to $0.53cm$. However, a 25% noise level is a very high level of noise. In practice, the intrinsic parameters are usually known with an accuracy that is better than 20%.

Tests on real images have confirmed what was suggested by the tests on simulated data. Table 4 shows in particular the importance of subpixel accuracy. Interest points extracted from the pattern's images have a 0.1-pixel precision while interest points extracted from the house's images have a 1-pixel (or more) precision. This difference in pixel precision between the two scenes explains the differences in average errors we obtained. Table 4 has confirmed once again that errors, up to 20%, on the intrinsic parameters do not have a significant effect on the accuracy of the 3D reconstruction, especially, the relative one. Furthermore, by looking more closely to Table 4, one can see that the aspect ratio has a greater effect on the accuracy of the reconstruction. For example, by comparing rows 2,3 and 6 to rows 4 and 5, we clearly see that when the value of the aspect ratio is accurate (close to the value from the calibration process), we obtain better results than those with a less accurate aspect ratio. This is a good news since the value of the aspect ratio is very stable and does not change with the camera settings, including focal-length changes [10]. Therefore, for a given camera, one should accurately estimate this aspect ratio only once.

Note that the normalization process is straightforward with a nonsignificant CPU-time overhead. Hence, the improved eight-point algorithm for calculating the essential matrix should be used for 3D reconstruction.

6 Conclusion

This paper has revisited the eight-point algorithm for the Euclidean three-dimensional reconstruction using uncalibrated images. Because approximate values of the camera's intrinsic parameters are usually available, either from the manufacturer's data or from previous experiment, we aimed at investigating the influence of noise, from the image coordinates (pixels) and from the intrinsic parameters' values, on the accuracy of the 3D reconstruction. The improved version of the classic eight-point algorithm was obtained by normalizing the pixel coordinates prior to the calculation of the essential matrix. The experiments have clearly shown the improvement of the accuracy of the recovered 3D structure. In particular, the accuracy of the relative 3D structure was excellent (average error was less than 1%) even when the approximate values of the intrinsic parameters were at about 20% off their exact values and pixel coordinates had a 1-pixel error. Although errors on the absolute 3D structure were slightly more important, such errors do not affect the 3D structure of the reconstructed scene and are only important for absolute measurements which are not always needed. The experiments have also uncovered that pixel noise has greater effects on the accuracy of the 3D reconstruction than noise on the intrinsic parameters. One can note that because approximate values for the intrinsic parameters might be at most 20% off the exact values, their effect will be offset by a 1-pixel noise on the coordinates as shown by our experiments. As a consequence, a calibration-based (or self-calibration) reconstruction method will not yield better results than the improved eight-point algorithm when the noise magnitude in the image coordinates is not less than one pixel. In other words, when the intrinsic parameters are known with a reasonable accuracy, the improved eight-point algorithm is a good choice for recover-

ing the relative Euclidean 3D structure. In addition, this algorithm has several advantages, it is linear, straightforward and easy to implement. Thus, practically speaking this algorithm might be preferred over other reconstruction methods, especially, the ones that require either explicit or self calibration.

References

[1] J Aisbett. An iterated estimation of the motion parameters of a rigid body from noisy displacement vectors. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1092–1098, 1990.

[2] B. Boufama and R. Mohr. A stable and accurate algorithm for computing epipolar geometry. *International Journal of Pattern Recognition and Artificial Intelligence*, 12(6):817–840, 1998.

[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] A. Fusiello. Uncalibrated euclidean reconstruction: a review. *Image and Vision Computing*, 18(2):555–563, 2000.

[6] R. Hartley. In defence of the eight-point algorithm. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[7] 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.

[8] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 579–587. Springer-Verlag, 1992.

[9] T.S. Huang and A.N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the* IEEE, 82(2):252–268, February 1994.

[10] J.M. Lavest, G. Rives, and M. Dhome. Utilisation d'un objectif à focale variable en vision monoculaire en vue de la reconstruction 3D. In *Actes du 8ème Congrès AFCET de Reconnaissance des Formes et Intelligence Artificielle, Lyon – Villeurbanne, France*, volume 1, pages 293–301. AFCET, November 1991.

[11] 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.

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

[13] A. Shashua. Projective structure from two uncalibrated images : Structure from motion and recognition. Technical Report A.I. Memo No. 1363, Massachusetts Institute of Technology, September 1992.

[14] M. E. Spetsakis and Y. Aloimonos. Optimal computing of structure from motion using point correspondences in two frames. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, California, USA*, pages 449–453, 1989.

[15] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

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

[17] J. Weng, N. Ahuja, and T.S. Huang. Optimal motion and structure estimation. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 15(9):864–884, September 1993.