Puerto Rico, June 17-19, 1997

IEEE Conference on Computer Vision and Pattern Recognition, pages 231-236, 1997

Self-Maintaining Camera Calibration Over Time*

Zhengyou Zhang[†] [‡] Veit Schenk[‡]
[†] ATR HIP, 2-2 Hikaridai, Seika-cho Soraku-gun, Kyoto 619-02 Japan
[‡] INRIA, 2004 route des Lucioles, BP 93, F-06902 Sophia-Antipolis Cedex, France
E-mail: zzhang@hip.atr.co.jp, zzhang@sophia.inria.fr

Abstract

The success of an intelligent robotic system depends on the performance of its vision-system which in turn depends to a great extend upon the quality of its calibration. During the execution of a task the vision-system is subject to external influences such as vibrations, thermal expansion etc. which affect and possibly render invalid the initial calibration. Moreover, it is possible that the parameters of the vision-system like e.g. the zoom or the focus are altered intentionally in order to perform specific vision-tasks. This paper describes a technique for automatically maintaining calibration of stereovision systems over time without using again any particular calibration apparatus. It uses all available information, i.e. both spatial and temporal data. Uncertainty is systematically manipulated and maintained. Synthetical and real data are used to validate the proposed technique, and the results compare very favourably with those given by classical calibration methods.

Keywords: Camera calibration, Calibration maintaining, Dynamic vision, Pose determination, 3D vision.

1. Introduction

Calibrating a camera consists in determining the analytical relationship between the three-dimensional coordinates of a point and the two-dimensional coordinates of its image by the camera. Once a camera model is chosen, the calibration problem is to compute the particular numerical parameters for a given camera. The classical methods are model-based. They are based on the observation of an objet for which the three-dimensional coordinates of N reference points[1] $\mathtt{M}_i = [X_i, Y_i, Z_i]^T$ are known. The projections $\mathbf{m}_i$ of these points are measured in the image and yield pixel coordinates $\mathbf{m}_i = [u_i, v_i]^T$. The reference objects which are used are generally calibration grids composed of repeated patterns (circles or rectangles) chosen to define point of interest which can be measured with a very good precision. A review of the state-of-the-art for camera calibration can be found in [6, 1].

In real applications (e.g. space applications), during performance of visual tasks, the camera calibration may be no longer valid due to accidental changes of the camera parameters such as thermal and mechanical influences as well as intentional changes in camera parameters such as a change in zoom and focus. One solution would be to detect, during a visual task, whether the calibration is no more valid. If it is no more valid, we could again use the classical technique to re-calibrate the vision system by showing calibration apparatus (model). This solution is of course not satisfactory because we have to interrupt a task being executed. Another solution is the so-called self-calibration [2, 3], which uses projective constraints between images and only requires to establish image-point correspondence without using any calibration apparatus. Unfortunately, up to now, there does not yet exist a robust and fully automatic self-calibration technique, and the calibration apparatus cannot yet be thrown away.

The work described here assumes that a vision system was initially calibrated by using some classical calibration technique. The objective is to maintain camera calibration over time using sequences of images of the surrounding environment, i.e. without having to use again a calibration object of precisely known dimensions. A camera is modeled by a standard pinhole. The relationship between the coordinates of a 3D space point and those of its image point is described by a 3×4 perspective projection matrix $\mathbf{P}$, which is defined up to a scale factor. The proposed method can, however, easily include more camera parameters such as radial distortion coefficients.

2. Summary of the proposed technique

The basic idea of the method is to 'push forward' Euclidean structure of the scene previously seen by the cameras until time instant t_i to the next instant t_{i+1}. This structure together with the information extracted from the corre-

* This work was partly supported by the European Space Agency (ESA) project CALVIN and was done at INRIA, France.

[1] We use a typewriter type style to denote vectors if uppercase letters are concerned, such as for space points.

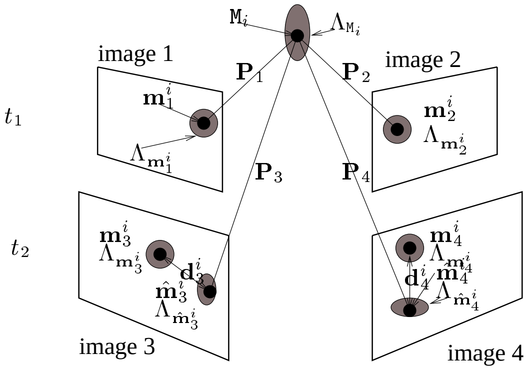

Figure 1. Notations used

$\mathtt{M}, \Lambda_{\mathtt{M}}$: 3D-points and their covariances, $\mathbf{m}, \Lambda_{\mathbf{m}}$: image-features and their covariances, $\hat{\mathbf{m}}, \Lambda_{\hat{\mathbf{m}}}$: reprojected points and their covariances, d: distance between observed image points and reprojected ones, $\mathbf{P}$: projection matrices

sponding images at t_{i+1} is used to obtain the new projection matrices and thus the internal and external parameters of the cameras with which the images have been taken. This is possible even when the camera parameters have changed in between views. The notations used are shown in figure 1. When image points are concerned, the subscript is used to denote the image number, and the superscript is used to denote the point number.

We now consider a binocular stereo system. (However, the method can be easily extended to deal with n cameras.) At each time instant, we consider two image pairs. For simplicity, let $i = 1$, but actually the technique described below will work with any i. The proposed method starts with strongly calibrated cameras at time t_1. We will assume that at t_1 the following is known:

- the projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ and their covariance matrices $\Lambda_{\mathbf{P}_1}$ and $\Lambda_{\mathbf{P}_2}$. They are provided by the classical calibration technique at the beginning of the session, and are then updated continuously by using the observed data with the technique being described.
- the point-matches $\mathbf{m}_{1-4}$ in images 1-4 with their respective covariance matrices $\Lambda_{\mathbf{m}_{1-4}}$. Usually, $\Lambda_{\mathbf{m}_i} = \Lambda_{\mathbf{m}_j}, \quad \forall_{i,j}$ because the images are taken with the same camera(s) or the same type of cameras and the same technique is used to extract the points of interest from images.

Using $\mathbf{P}_1$, $\mathbf{P}_2$, $\Lambda_{\mathbf{P}_1}$, $\Lambda_{\mathbf{P}_2}$, $\mathbf{m}_{1,2}$ and $\Lambda_{\mathbf{m}_{1,2}}$, a set of 3D points $\mathtt{M}_i$ and their covariance matrices $\Lambda_{\mathtt{M}_i}$ can be computed (Sect. 3.1). Using the 3D reconstruction $\mathtt{M}_i$ and $\Lambda_{\mathtt{M}_i}$ and the image points $\mathbf{m}_{3,4}$ together with $\Lambda_{\mathbf{m}_{3,4}}$, the projection matrices $\mathbf{P}_3$ and $\mathbf{P}_4$ and their covariance matrices $\Lambda_{\mathbf{P}_3}$ and $\Lambda_{\mathbf{P}_4}$ at time instant t_2 can be obtained (Sect. 3.2).

3. Description of the Proposed Technique

This section provides the details of the proposed technique. The first step is to reconstruct the 3D points observed at t_1 together with their uncertainty measure. The second step is then to update the camera projection matrices by minimizing the error between the observed image points at t_2 and the projection of the 3D points reconstructed at t_1. One particular effort is on the characterization of the uncertainty. This is important because the errors of the reconstructed points are not the same in different directions (e.g. there is usually a larger error in depth than in lateral directions) and they are different from one point to another [4, 7].

3.1. Calculating the 3D reconstruction

We are given at t_1 point matches $\{(\mathbf{m}_1^i, \mathbf{m}_2^i)\}$, their covariance matrices $\{(\Lambda_{\mathbf{m}_1^i}, \Lambda_{\mathbf{m}_2^i})\}$, and the projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ with covariance matrices $\Lambda_{\mathbf{P}_1}$ and $\Lambda_{\mathbf{P}_2}$.

Analytical solution: For each pair of image-correspondences, we obtain, from the pinhole camera model, 4 linear equations in the 3 unknown coordinates. A linear least-squares technique is used to compute the 3 unknowns. Details are omitted here.

Nonlinear refinement: The above solution is not optimal because the quantity being minimized does not have a physically meaningful interpretation. Analysis based on maximum likelihood principle shows that we should minimize the Mahalanobis distance in the image-plane between observed image points $\mathbf{m}_i$ and reprojected points $\hat{\mathbf{m}}_i$:

$$\min_{\mathtt{M}} \sum_{j=1}^{2} \mathbf{d}_j^T \Lambda_j^{-1} \mathbf{d}_j \tag{1}$$

where $\mathbf{d}_j = (\mathbf{m}_j - \hat{\mathbf{m}}_j)$, the reprojected point is obtained based on the camera model, i.e. $\hat{\mathbf{m}}_j = \mathbf{P}_j(\mathtt{M})$, and Λ_j is defined as $\Lambda_j = \Lambda_{\mathbf{m}_j} + \Lambda_{\hat{\mathbf{m}}_j}$ where $\Lambda_{\mathbf{m}_j}$ is the uncertainty in image j (which is fixed by the user or obtained through the analysis of the corner detector used) and $\Lambda_{\hat{\mathbf{m}}_j}$ is the uncertainty of the reprojected point due to the uncertainty of the camera projection matrices, defined as

$$\Lambda_{\hat{\mathbf{m}}_j} = \frac{\partial \hat{\mathbf{m}}_j}{\partial \mathtt{P}_j} \Lambda_{\mathtt{P}_j} \left(\frac{\partial \hat{\mathbf{m}}_j}{\partial \mathtt{P}_j} \right)^T \tag{2}$$

The term $\Lambda_{\mathtt{P}_j}$ is the uncertainty of the projection matrix $\mathbf{P}_j$ which at t_1 is given by the classical camera calibration technique. Starting from t_2 these uncertainty-matrices are the result of the calculations in section 3.2. Note that in the above computation, a projection matrix $\mathbf{P}$ is used as a 12-D vector $\mathtt{P}$, and $\Lambda_{\mathtt{P}_i}$ is considered to be a 12x12 matrix. The minimization of (1) is conducted using the Levenberg-Marquardt technique.

Covariance for 3D points: Now we show how to estimate the covariance-matrices of the estimated 3D points. The 3D points are obtained by solving the minimization problem (1), that is by minimizing

$$C(\mathtt{M},\mathbf{x}) = \sum_{j=1}^{2}(\mathbf{m}_j - \hat{\mathbf{m}}_j)^T \boldsymbol{\Lambda}_i^{-1}(\mathbf{m}_j - \hat{\mathbf{m}}_j) \ . \qquad (3)$$

Here, we define the measurement vector $\mathbf{x}$ to be the 28-vector $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{P}_1, \mathbf{P}_2)$ (here again, we consider the camera projection matrix as a 12-vector). Using the implicit function theorem, we can compute the covariance matrix of $\mathtt{M}$ as

$$\Lambda_{\mathtt{M}} = \mathbf{D}\Lambda_{\mathbf{x}}\mathbf{D}^T \quad \text{with } \mathbf{D} = \left[\frac{\partial^2 C(\hat{\mathtt{M}}, \hat{\mathbf{x}})}{\partial \mathtt{M}^2}\right]^{-1} \frac{\partial^2 C(\hat{\mathtt{M}}, \hat{\mathbf{x}})}{\partial \mathtt{M} \partial \mathbf{x}} \ .$$

Here, the covariance matrix of $\mathbf{x}$ is given by $\Lambda_{\mathbf{x}} = \text{diag}\,(\boldsymbol{\Lambda}_{\mathbf{m}_1}, \boldsymbol{\Lambda}_{\mathbf{m}_2}, \boldsymbol{\Lambda}_{\mathtt{P}_1}, \boldsymbol{\Lambda}_{\mathtt{P}_2})$. Here again, we consider $\boldsymbol{\Lambda}_{\mathtt{P}_i}$ ($i = 1, 2$) to be a 12×12 matrix.

3.2. Estimating $\mathbf{P}$

We now describe how to estimate the projection matrices at t_2, $\mathbf{P}_i$ ($i = 3, 4$), and their covariance matrices $\Lambda_{\mathbf{P}_i}$ ($i = 3, 4$). Since the technique described here works in exactly the same way for both projection matrices, we only consider $\mathbf{P}_3$ in the sequel.

We have a set of Euclidean points $\mathtt{M}_i$ ($i = 1, \dots, n$) reconstructed from previously observed image points together with their covariance matrices $\Lambda_{\mathtt{M}_i}$ ($i = 1, \dots, n$), as described in section 3.1. We are also given the correspondence between these 3D points $\mathtt{M}_i$ and the image points $\mathbf{m}_3^i$ observed in image 3. The projection matrix $\mathbf{P}_3$ is obtained by minimizing the distance between the image points $\mathbf{m}_3^i$ and the reprojected points $\hat{\mathbf{m}}_3^i$ (reprojected from the 3D-reconstruction obtained previously). This minimization is done using the Levenberg-Marquardt least-squares technique which requires an initial guess.

Linear method for finding initial estimate for $\mathbf{P}$: In order to find an initial estimate the uncertainty measures are ignored and only the algebraic distances are used. Starting with the classical pinhole model equation

$$s[u, v, 1]^T = \mathbf{P}[X, Y, Z, 1]^T \ ,$$

where (u, v) are 2D coordinates and (X, Y, Z) are 3D coordinates, we eliminate the factor s and create a matrix $\mathbf{A}$ with 2 rows per point-correspondence, which yields

$$\mathbf{A}\mathbf{x} = \mathbf{0}$$

where $\mathbf{x}$ is a 12-vector composed by the entries of camera perspective projection matrix $\mathbf{P}_3$. Since $\mathbf{P}_3$ is defined up to a scale factor, we can impose the constraint that $\|\mathbf{x}\| = 1$. The solution to the above problem is simply the eigenvector of $\mathbf{A}^T\mathbf{A}$ associated with the smallest eigenvalue.

Nonlinear refinement of the initial guess. Start from the above initial estimate, we refine the projection matrix by minimizing the Mahalanobis distance in the image-plane between the image-points $\mathbf{m}_3^i$ and the reprojected points $\hat{\mathbf{m}}_3^i$:

$$\min_{\mathbf{P}} \sum_i \mathbf{d}_i^T \Lambda_i^{-1} \mathbf{d}_i \qquad (4)$$

where $\mathbf{d}_i = \mathbf{m}_3^i - \hat{\mathbf{m}}_3^i$, $\hat{\mathbf{m}}_3^i = \mathbf{P}_3(\mathtt{M}_i)$, Λ_i is defined as $\Lambda_i = \Lambda_{\mathbf{m}_3^i} + \Lambda_{\hat{\mathbf{m}}_3^i}$ where $\Lambda_{\mathbf{m}_3^i}$ is the covariance matrix of point i in image 3 and $\Lambda_{\hat{\mathbf{m}}_3^i}$ is defined as

$$\Lambda_{\hat{\mathbf{m}}_3^i} = \frac{\partial \hat{\mathbf{m}}_3^i}{\partial \mathtt{M}_i} \Lambda_{\mathtt{M}_i} \left(\frac{\partial \hat{\mathbf{m}}_3^i}{\partial \mathtt{M}_i}\right)^T \qquad (5)$$

which is the uncertainty of the reprojected points. The term $\Lambda_{\mathtt{M}_i}$ is the covariance matrix of the 3D points calculated in section 3.1.

The minimization is done using the Levenberg-Marquardt technique. Since $\mathbf{P}_3$ is defined up to a scale factor, we need to appropriately parameterize it. One way is to set the element of the last row having the largest value to 1 and use the remaining 11 elements as free parameters.

Covariance matrix for $\mathbf{P}$. Exactly the same algebra as for the covariance matrix of reconstructed 3D points can be done for that of the camera projection matrix $\mathbf{P}$. We only need to mention that the covariance matrix is a function of $\mathbf{m}_3^i$, $\mathtt{M}_i$, and their covariance matrices.

4. Experimental Results

Two sets of experiments were performed, the first one using synthetic data in order to investigate the robustness of the algorithms in the presence of noise, while the second one using real image-data to confront our proposed technique with the real world.

4.1. Synthetic Data

In the experiments involving synthetic data, the following was used:

- the known 3D coordinates of an object, in this case the calibration grid known as mire44 as shown in figure 2.
- four different projection matrices (with fixed, i.e. known parameters). The exact values used are:

$\mathbf{P}$	α_u	α_v	u_0	v_0	c
1	600	800	250	260	-4.62593e-20
2	560	750	255	265	2.47818e-20
3	650	830	255	265	2.05067e-12
4	600	800	260	270	7.19234e-11

  which corresponds to something like a small zoom between t_1 and t_2. The parameter c above is equal to $-\alpha_u \cot\theta$, which is very close to 0 because θ is very close to $\pi/2$. The external parameters are:

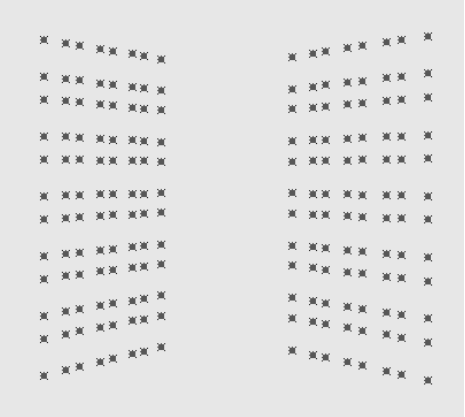

Figure 2. Synthetic Data: The Calibration-Object ‘mire44’

$\mathbf{P}$	r_x	r_y	r_z	t_x	t_y	t_z
1	-2.61e-16	5.498	1.08e-16	-200	-150	1000
2	-2.61e-16	5.498	1.08e-16	-323	-150	1030
3	4.49e-09	5.273	8.12e-09	-171	-110	982
4	1.52e-07	5.273	2.75e-07	-294	-110	1012

which corresponds to a slight translation of the cameras and a rotation around the y-axis. The translation vector corresponds to the position of the optical center in the absolute coordinate system. The rotation is represented as the rotation-axis with the norm of the rotation-vector being the angle of the rotation.

- the four views generated from the 3D points projected by the projection matrices.

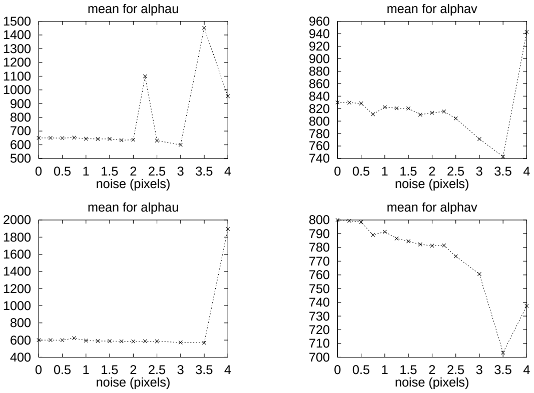

Figure 3. Behavior of Algorithm in the Presence of Noise: α_u and α_v for images 3 (top) and 4 (bottom)

The results for the experiments using synthetic data are shown in figures 3 - 6. The units for t_x, t_y and t_z are millimeters, while those for r_x, r_y and r_z, in radians. The results are very good up to a noise with standard deviation of about 2 pixels per point. The estimation of the intrinsic parameters is stable. The relative error in translation is about 3%, while that in rotation is about 2%. At noise-level of above approximately 2 pixels per image-point, the results become less useful. One reason is that the 3D points used for calibration are reconstructed from 2D noisy points at t_1 and that the 2D points at t_2 used for calibration are also noisy, which is different from the classical calibration where the 3D points are known very precisely.

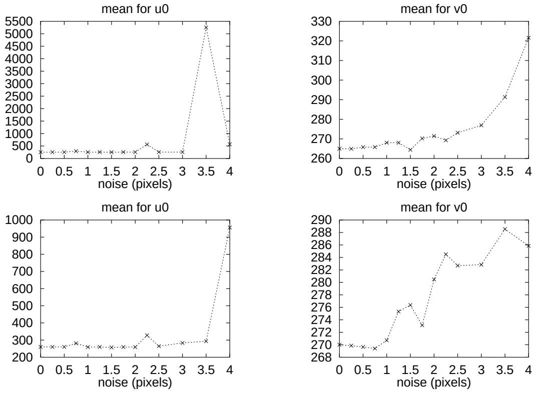

Figure 4. Behavior of Algorithm in the Presence of Noise: u_0 and v_0 for images 3 (top) and 4 (bottom)

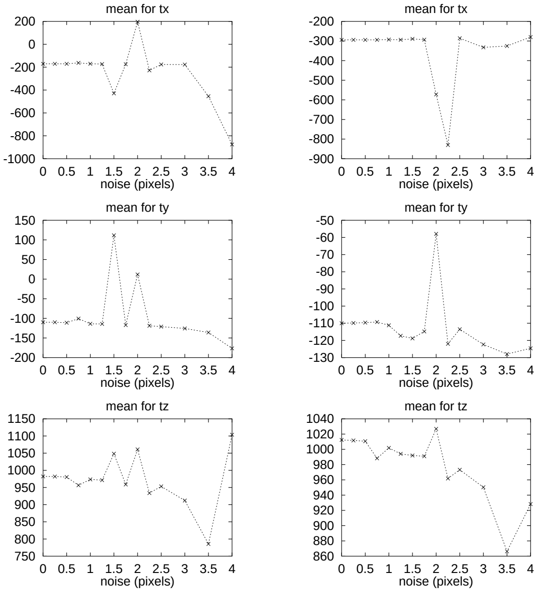

Figure 5. Behavior of Algorithm in the Presence of Noise: t_x, t_y and t_z for images 3 (left) and 4 (right)

4.2. Real data

For the experiments involving real images a short sequence of the same scene was taken: the first pair corre-

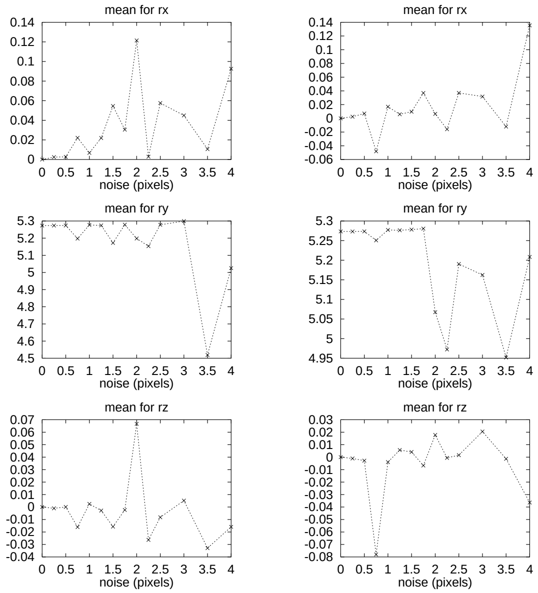

Figure 6. Behavior of Algorithm in the Presence of Noise: r_x, r_y and r_z for images 3 (left) and 4 (right)

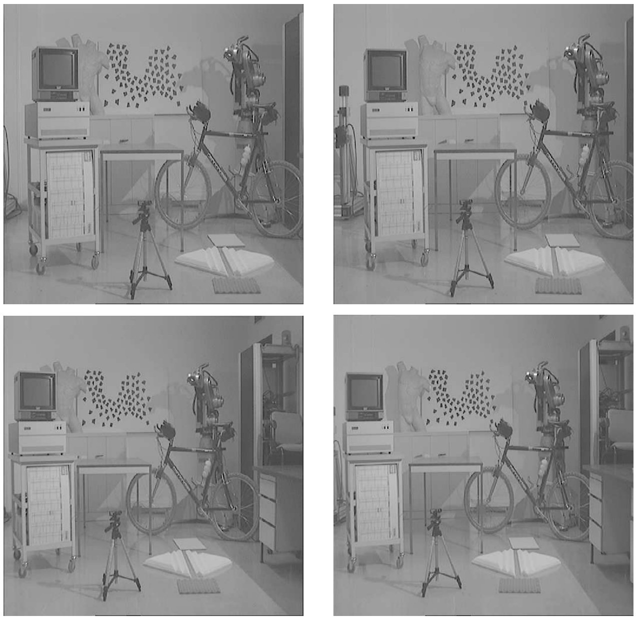

Figure 7. Real image data: Top: images at t_1, Bottom: images at t_2

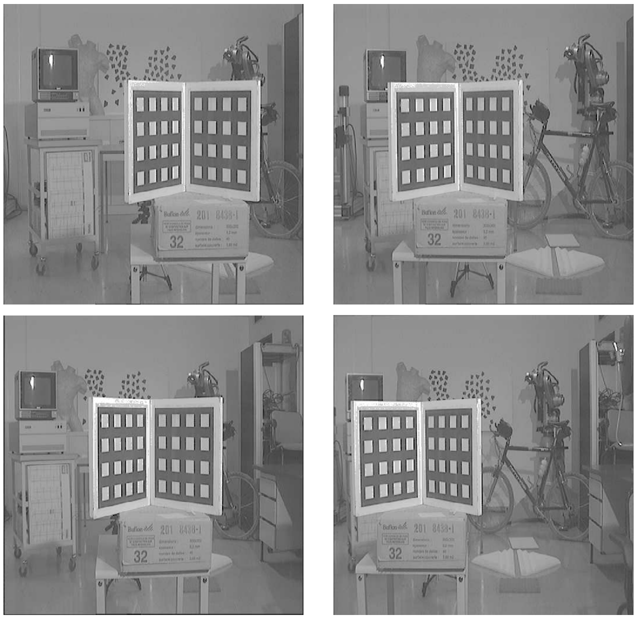

Figure 8. Images taken with the same configuration as in figure 7 but with calibration object placed in front of camera

sponding to a stereo-head at instant t_1 the second pair at t_2. Between t_1 and t_2 the stereo-head has performed a translation towards the scene and a small rotation to the right, and has decreased the zoom in order to keep the objects of interest at approximately the same size. These images are shown in figure 7. In order to be able to compare the obtained results with those of classical calibration techniques, four images were taken with the same configuration but with a calibration object placed in front of the camera. These images are shown in figure 8. Since the calibration object was place in exactly the same position for all for shots, it appears in different positions of the images. The points of interest were extracted from the images in question and matched across all 4 images giving a total of about 200 common point matches in the images. This was done using the `image-matching` software. The left image at instant t_2 with the matched feature points marked is shown in figure 9.

The results for the experiments involving real data are shown in table 1. $\mathbf{P}_i$ $(i = 1, \ldots, 4)$ are given by a classical calibration technique [5], while $\widehat{\mathbf{P}}_j$ $(j = 3, 4)$ are obtained with the technique proposed in this paper. The results for both the intrinsic as well the extrinsic parameters are very close to the ones given by the classical method, except for the fourth image. In fact, the results for the fourth image obtained with our new technique are closer to what one would expect from the experimental setup than the results given by a classical calibration technique, because between the two images at t_2 the same camera was translated and rotated, but the intrinsic parameters remained unchanged. The classical method gives different intrinsic parameters for the two images (e.g., α_u and α_v are around 1200 for $\mathbf{P}_3$, but 1100 for $\mathbf{P}_4$), the new method on the other hand gives almost identical values for α_u and α_v (they are all very close to 1200 for $\widehat{\mathbf{P}}_3$ and $\widehat{\mathbf{P}}_4$). The different values given by the classical technique could be due to the fact that the calibration object in the fourth image is in the left part of the image (see bot-

Table 1. Results for Real Data: Values at t_1 are given in lines $\mathbf{P}_1$ and $\mathbf{P}_2$. Lines $\mathbf{P}_3$ and $\mathbf{P}_4$ give the results of the classical method, the lines $\widehat{\mathbf{P}}_3$ and $\widehat{\mathbf{P}}_4$ give the results of the new method

	α_u	α_v	u_0	v_0	$\mathbf{r}$	$\mathbf{t}$
$\mathbf{P}_1$	1465	1466	380	316	$[-0.576, 5.39, 0.278]$	$[-1482, -10.9, -1372]$
$\mathbf{P}_2$	1393	1391	310	274	$[-0.465, 5.55, 0.292]$	$[-1126, -0.24, -1634]$
$\mathbf{P}_3$	1212	1211	396	314	$[-0.538, 5.33, 0.274]$	$[-1212, -2.09 - 1181]$
$\widehat{\mathbf{P}}_3$	1220	1208	311	308	$[-0.549, 5.40, 0.297]$	$[-1216, -0.24, -1198]$
$\mathbf{P}_4$	1102	1105	268	256	$[-0.327, 5.55, 0.270]$	$[-849, -10.0, -1406]$
$\widehat{\mathbf{P}}_4$	1190	1203	304	280	$[-0.480, 5.51, 0.296]$	$[-955, -3.77, -1535]$

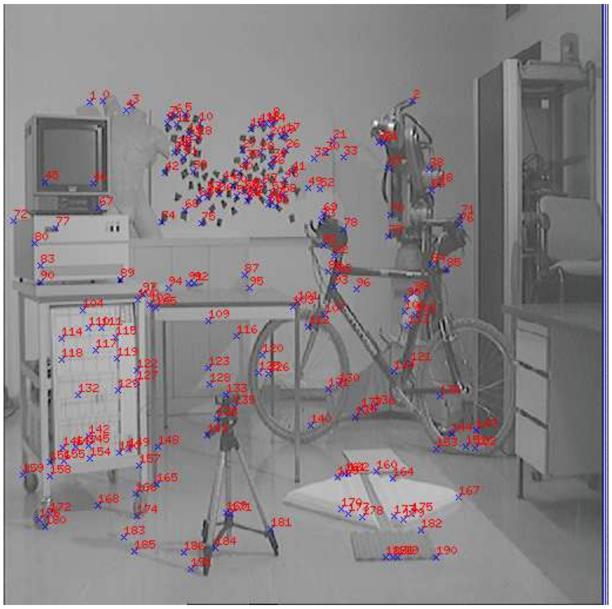

Figure 9. Left image at t_2 from figure 7 shown with matched points common to all 4 images

tom right of figure 8) and not near the center of the image. It is a well-known fact that the classical calibration technique is very sensitive to the size and position occupied by the calibration pattern in the image.

5. Conclusion

A new method has been proposed for obtaining camera-calibration of a stereovision system over time without using again any particular calibration apparatus. The idea is to use previously valid camera projection matrices and image point matches to push forward the Euclidean structure of the scene, which allows us to recalibrate the stereovision system. Uncertainty is systematically manipulated and maintained. This is important because the errors of the reconstructed points are different in different directions and from one point to another. This, together with the configuration of observed image points, affects the precision of the estimated camera projection matrices. They cannot be properly and efficiently propagated over time without correctly characterizing their uncertainty. The proposed method has been evaluated using both synthetic data with various levels of noise added as well as image-data obtained using real cameras. The results compare very favorably with those given by classical calibration-methods.

In the current work, no knowledge of the vision system is assumed, i.e. all parameters of the cameras are free to change. This is usually not the case in practice (e.g., only zoom is modified). Our future work will be the development of a technique which tracks the calibration parameters by taking into account the knowledge of their variation.

References

[1] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[2] O. Faugeras, T. Luong, and S. Maybank. Camera self-calibration: theory and experiments. In G. Sandini, editor, *Proc 2nd ECCV*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[3] Q.-T. Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systèmes robotiques*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, Dec. 1992.

[4] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Transactions on Robotics and Automation*, 3:239–248, 1987.

[5] L. Robert. Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing*, 63(2):314–325, Mar. 1995. also INRIA Technical Report 2204.

[6] R. Tsai. Synopsis of recent progress on camera calibration for 3D machine vision. In O. Khatib, J. J. Craig, and T. Lozano-Pérez, editors, *The Robotics Review*, pages 147–159. MIT Press, 1989.

[7] Z. Zhang and O. D. Faugeras. *3D Dynamic Scene Analysis: A Stereo Based Approach*. Springer, Berlin, Heidelberg, 1992.