VIDEO SEQUENCE CALIBRATION AND APPLICATIONS

L. ÁLVAREZ, C. CUENCA, A.SALGADO AND J.SANCHEZ
Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria
Campus Universitario de Tafira
35017 Las Palmas. SPAIN
email: {lalvarez, ccuenca, jsanchez}@dis.ulpgc.es
www: http://serdis.dis.ulpgc.es/ami

ABSTRACT

In this paper we propose a new technique to calibrate a video sequence. We will compute the location and rotation of the camera for each frame of the video sequence. Classical methods for camera calibration work properly in the case of a small number of cameras. When we take an image sequence with a video rate, the number of frames is very large and classical methods do not work properly. To avoid this drawback, we focus the calibration process in the estimation of the coordinates of a set of $3D$ points in the scene, so the unknown are these 3D point coordinates rather than the projection matrix associated to each frame. The advantage of this approach is that with a few number of 3D point coordinates we can estimate the projection matrix of any frame using the relation between the 3D points and its projection on each frame. In order to compute such set of 3D points, we will use a classical camera calibration technique applied on a small subsequence of frames taken from the large original video sequence. We also perform an iterative procedure to include the information of all the cameras in the calibration computation. To illustrate the capabilities of the proposed method, we will apply this technique to include virtual 3D objects in a real video sequence.

KEYWORDS

Augmented Reality, Geometric Algorithms, Tracking, Motion Control.

1. Introduction

The mathematical aspects of multiple camera calibration have been deeply studied in the literature [3],[4]. In the case of a few number of cameras the classical calibration techniques work properly. However to calibrate a video sequence we have to deal with two important problems: On one hand we have to calibrate a large number of cameras (one camera per video frame) and on the other hand the displacement between consecutive frames is in general very small and the recovering of the camera parameters based on the tracking of singular points across the sequence is very unstable with a lot of local minima configuration far away from the physical relevant solution. There are sophisticated techniques to deal with these problems; however, probably due to the commercial interest of such techniques, there is not much public information about specific tools that deal with these problems. In this paper we present a method to calibrate a video sequence properly. We do not claim that this method is better than the sophisticated tools presented in some commercial software packages (In particular we assume that the intrinsic parameters of the camera are known, which is a restriction that could be avoided). However, the technique we propose in this paper seems to work properly, as it is shown in the experimental results, and it could be easily implemented for any domain researcher.

The remainder of this paper is organized as follows: In section 2 we present a general overview of multiple camera calibration techniques. In section 3 we present the method we propose in this paper. In section 4, we present an application of this technique: the inclusion of virtual objets in a real video sequence. Finally, in section 5 we present the main conclusions of this paper.

2. Multiple Camera Calibration. A General Overview

The problem of multiple camera calibration consists in recovering the camera positions and orientations with respect to the world coordinate system, using as input data tokens, such as pixels or lines, in correspondence in different images. Figure 1 shows this scenario for a system with three cameras.

The specification of the i-th camera position is the 3D point $C_i^{(world)}$, where the superscript is the reference system in which the magnitude is expressed. The orientation specification is a rotation matrix $R_i^{(world)}$ or any equivalent representation, such as quaternions or Euler angles.

When the image tokens in correspondence are projections of a set of 3D points $\{M_j\}_{j=1..N}$, where N is the number of points, it is possible to reconstruct each 3D point

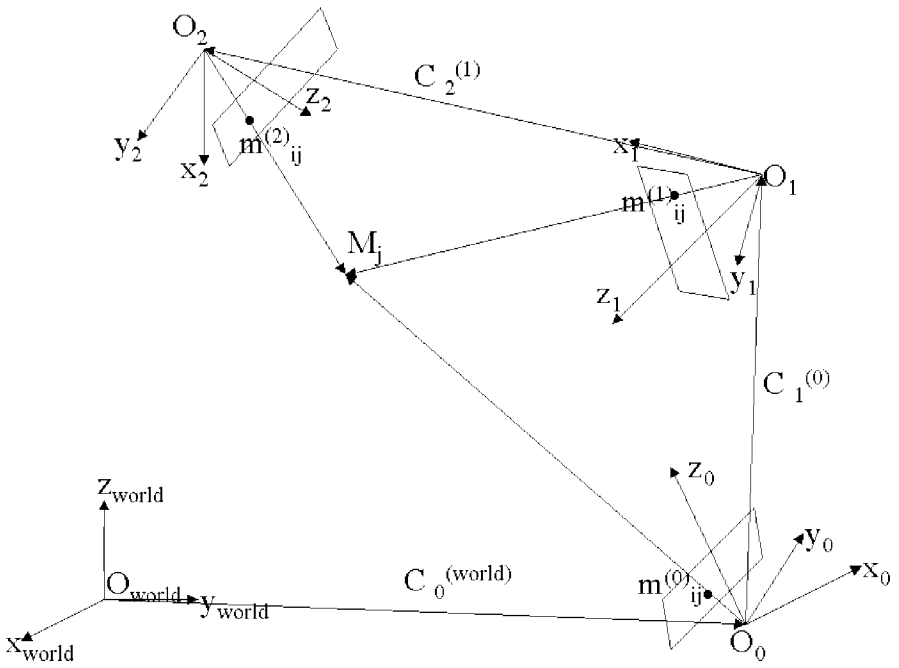

Figure 1. Motion parameters derived from point matches.

expressed in the world coordinate system by simply estimating the intersection point of the line set:

$$\left\{ r_i \equiv C_i^{(world)} + \lambda \overrightarrow{C_i^{(world)} R_i^{(world)} m_{ij}^{(i)}} \right\}_{i=1..N}$$

where $C_i^{(world)}$ are the coordinates of the optical center in the world reference system, and $m_{ij}^{(i)}$ are the coordinates of the projection of M_j in the normalized reference system for the i-th camera. A reference system is normalized when the optical center is in the origin, the focal distance is 1 and the pixel is a square of size 1. We will assume that the intrinsic parameters of the cameras are known, which allows us to normalize the reference system.

In order to estimate the intersection 3D point of the line set it is necessary to know the position of the optical center and the rotation matrix for each camera. The computation of these parameters solves the problem of the multiple camera calibration. After estimating these parameters, we can evaluate the solution accuracy by projecting the reconstructed 3D points in each camera, and the best solution for the calibration problem is the one that minimizes the energy function:

$$f(C_0^{(world)}, C_1^{(world)} \quad , \quad ..., R_0^{(world)}, R_1^{(world)} ...) = \sum_{i,j} \left\| m_{ij}^{\prime(i)} - m_{ij}^{(i)} \right\|^2$$

where $m_{ij}^{\prime(i)}$ is the projection of the reconstructed point M_j in the i-th camera

There is no closed-form solution for the minimization of the above energy fuction, and nonlinear minimization methods must be used.

A restriction to take into account in the application of these methods is that the solution must be not only minimum but also valid. (A solution is valid when firstly the rotation matrixes are orthogonal, and secondly, the reconstructed 3D points are always beyond the image plane, since the reference system in the camera is normalized.)

It is important to find a good initial approximation, close to the final solution, in order to supply as seed input to the nonlinear minimization method which guarantees a fast convergence.

This initial solution can be obtained by using linear methods. In [2], the *essential matrix* E (for motion parameters (t, R)) is defined, by:

$$E = TR \tag{1}$$

where T is the antisymmetric matrix:

$$\begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{2}$$

Matrix T is such that $Tx = t \wedge x$ for all vectors x. The nine elements E_{ij} of E are called *essential parameters*. Since two sets of points $\{m_j\}$ and $\{m'_j\}, i = 1, ..., N$ can be interpreted as resulting from a 3D camera motion (t, R) if and only if $\mid t, m_j, Rm'_j \mid = 0$, then the last equation can be written as shown in [5]:

$$m_j^T E m_j' = 0 \tag{3}$$

Equation (3) is linear and homogeneous with respect to the coefficient of E. Therefore, if eight of such equations are available, we can solve the system for those coefficients. The approach consists in first, estimating the essential matrix E, and then recovering t and R from E. It is possible to rewrite equation (3) as:

$$a^T X = 0 \tag{4}$$

where X is the 9x1 vector $[e_1^T, e_2^T, e_3^T]^T$ (e_i is the i-th column vector of the essential matrix E), and a is the 9x1 vector $[xm'^T ym'^T zm'^T]^T$.

If we have n points in correspondence, each one yields an equation like (4) and we can combine them as follows:

$$A_n X = 0 \tag{5}$$

where A_n is an nx9 matrix:

$$A_n = \begin{pmatrix} a_1^T \\ \vdots \\ a_n^T \end{pmatrix} \tag{6}$$

In the presence of noise, equation (5) is only approximately satisfied, and we can reformulate the problem as that of finding the vector X that minimizes the norm of $A_n X$ with the constraint that the norm of X is $\sqrt{2}$.

It is well known that the solution to the last problem is the eigenvector of norm $\sqrt{2}$ of the $9x9$ matrix $A_n^T A_n$ corresponding to the smallest eigenvalue.

The computation of t and R can also be performed taking noise into account. The translation vector t is the solution of the following meansquare problem:

$$min_t ||E^T t||^2 \tag{7}$$

with the constraint that the norm of $||t||^2 = 1$

In order to find the rotation matrix R, we have to solve the following meansquare problem:

$$min_R ||E - TR||^2 \tag{8}$$

subject to $R^T R = I$ and $|R| = 1$

With this method it is possible to calibrate a system with two cameras and without noise. When noise is present, the method must be slightly changed because it is not possible to find a valid solution to the calibration problem. Theses changes consist simply in introducing heuristical rules to select the *best* solution.

To extend the method to more than two cameras it is enough to carry out the calibration for each couple of cameras (the first camera and the second, the second and the third, and so on) and to fit a scale factor for each couple of cameras.

In order to calculate the scale factor of two pairs of cameras, we reconstruct each 3D point, M_j, from its respective correspondence pairs and then we minimize the expression:

$$\sum_j \left\| M_j^{(0)} - \overrightarrow{O_0 O_1} - \lambda R_1^{(0)} M_j^{(1)} \right\|$$

The analytical solution to this minimization problem is given by:

$$\lambda = \frac{\sum_j (M_j^{(0)} - \overrightarrow{O_0 O_1}) \cdot R_1^{(0)} M_j^{(1)}}{\sum_j \left\| M_j^{(1)} \right\|^2}$$

This method has the advantage of being linear and the disadvantage of being very noise sensitive (hence the importance of a good estimation of the point coordinates provided by a corner detection technique). Moreover, the method does not take advantage of having multiple cameras in order to improve the result of the calibration.

We can include all cameras using the linear solution as initial approximation for minimization of equation (1). In each iteration, we must take into account if the solution is still valid. A possible strategy consists in adding a heavy penalty term to the equation (1) when the restrictions are violated.

3. Video Sequence Calibration

The method we propose in this paper to calibrate a video sequence is divided into the following steps :

Step 1

In the first step we compute automatically sequences of corresponding singular points across the video sequence. We can use as singular points corners detected using the classical Harris technique or a more sophisticated one, as the one introduced in [1].

Step 2

In the second step we choose a small subsequence of frames to recover a set of 3D points. The selected frames could be chosen by hand, or taking a fixed step between frames (i.e. we take for instance frames 1-25-50-75-.....). We could also use a more sophisticated way to choose the frames based on the robustness of the calibration information between two cameras. Such robustness is based on the two smallest eigenvalues $\lambda_1 < \lambda_2$ of the matrix $A_n^T A_n$ (see (5)). The ideal case is that $\lambda_1 = 0$ and $\lambda_2 >> 0$. So we can choose the frames by maximizing some criteria associated to such robustness, for instance

$$\frac{\lambda_2 - \lambda_1}{\lambda_2 + \epsilon}$$

Once the video frame subsequence is obtained we compute a set of 3D points M_j using the technique showed in the previous section.

iteration	step 1	step 16
0	25	5.6
8	3.7	1.1
16	2.3	0.9
32	1.6	0.8
63	1.4	0.6

Table 1. Average reprojection error evolution

Step 3

First we notice that from the tracking step we know for each 3D point M_j the projection of the point $m_{ij}^{(i)}$ in the i-th camera. From these relations we can compute the projection matrix P_i associated to each camera (see [2] for details).

We update the 3D point coordinates and the projection matrix using the following iterative scheme:

- From M_j and $m_{ij}^{(i)}$ we compute the projection matrix P_i for all frames in the video sequence.
- From P_i and $m_{ij}^{(i)}$ we recompute the 3D points M_j by intercepting the 3D lines going from $m_{ij}^{(i)}$ to the focus of P_i.
- We update M_j with the new computed ones and we start a new iteration until convergence of the iterative scheme.

4. Experimental results. Inclusion of virtual objects in a real video sequence

One of the main applications of video sequence calibration is the inclusion of virtual objects in a real video sequence. We will test our method in a real video sequence of 120 frames where we are going to include four artificial objets. In figure 2 we present four frames of the real video sequence (left column) and the same frames with the inclusion of the virtual objets using the calibration parameters obtained with the proposed method (right column).

To illustrate the convergence behavior of the proposed iterative scheme we present in figure 3 the evolution across iterations of the average reprojection error in terms of image pixel values. We present the results using two different initial video subsequences. The first one is obtained using frames 1-17-33-49-... (that is, we fit an step of 16 frames). The second one is obtained using all frames (step equal to 1).

In table 1 we present the numerical values of the average reprojection error for the iterative scheme evolution presented in figure 3.

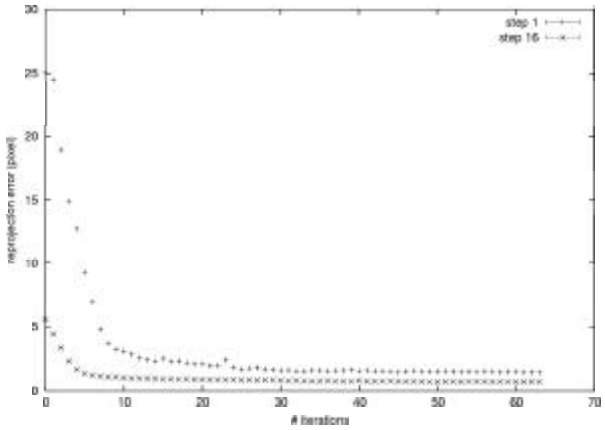

Figure 3. Average reprojection error evolution

5. Conclusions

In this paper we present a new method to perform video sequence calibration. The method is quite simple and seems to work properly. The experimental results are very promising, as well as the convergence behavior of the iterative scheme. In a real video sequence we arrive to get an average reprojection error of 0.6 pixels which is a very good estimation.

References

[1] Alvarez L., Cuenca C., Mazorra L. 2001. Morphological Corner Detection. Application to Camera Calibration, *Proceedings of IASTED International Conference SIGNAL PROCESSING, PATTERN RECOGNITION AND APPLICATIONS, Rhodes, Greece* , pages 21–26

[2] Faugeras O. 1993. *3D Computer Vision. A Geometric View Point*. MIT Press.

[3] Faugeras O., Luong Q-T, Papadopoulo T. 2001, *The Geometry of Multiple Images*, MIT Press

[4] Hartley R. and Zisserman A. 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[5] Kanatani K. 1995. *Geometric Computation for Machine Vision*. Oxford University Press.

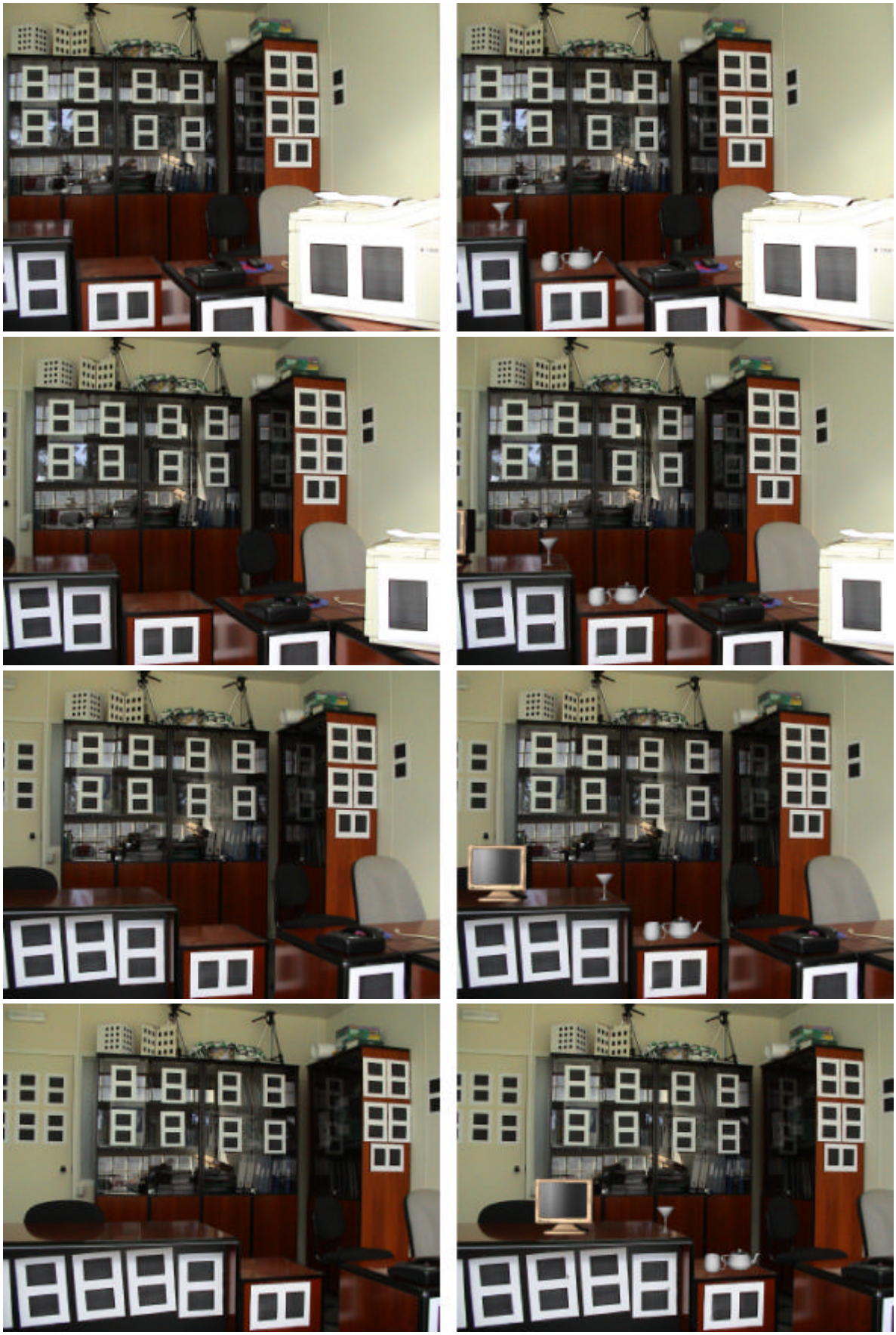

Figure 2. On the left, four frames of a real video sequence and on the right the same frames with the inclusion of the virtual objets (a glass, a mug, a teapot and a computer screen)