Registration with a Moving Zoom Lens Camera for Augmented Reality Applications

Gilles Simon and Marie-Odile Berger

LORIA- INRIA Lorraine
BP 101
54602 Villers les Nancy, France
email:{gsimon@loria.fr,berger@loria.fr}

Abstract. We focus in this paper on the problem of adding computer-generated objects in video sequences that have been shot with a zoom lens camera. While numerous papers have been devoted to registration with fixed focal length, little attention has been brought to zoom lens cameras. In this paper, we propose an efficient two-stage algorithm for handling zoom changing which are are likely to happen in a video sequence. We first attempt to partition the video into camera motions and zoom variations. Then, classical registration methods are used on the image frames labeled *camera motion* while keeping the internal parameters constant, whereas the zoom parameters are only updated for the frames labeled *zoom variations*. Results are presented demonstrating registration on various sequences. Augmented video sequences are also shown.

1 Introduction

Augmented Reality (AR) is a technique in which the user's view is enhanced or augmented with additional information generated from a computer model. In contrast to virtual reality, where the user is immersed in a completely computer-generated world, AR allows the user to interact with the real world in a natural way. This explains why interest in AR has substantially increased in the past few years and medical, manufacturing or urban planning applications have been developed [2,5,15,18].

In order to make AR systems effective, the computer generated objects and the real scene must be combined seamlessly so that the virtual objects align well with the real ones. It is therefore essential to determine accurately the location and the optical properties of the cameras. The registration task must be achieved with special care because the human visual system is very good at detecting even small mis-registrations.

There has been much research in the field of vision-based registration for augmented reality [1,12,14,18]. However these works assume that the internal parameters of the camera are known (focal length, aspect ratio, principal point) and they only address the problem of computing the pose of the camera. This is a strong limitation of these methods because zoom changing is likely to happen

D. Vernon (Ed.): ECCV 2000, LNCS 1843, pp. 578–594, 2000.

in a video sequence. A method is proposed in [11], which can retrieve metric reconstruction from image sequences obtained with uncalibrated zooming cameras. However, considering unknown principal point leads to unstable results if the projective calibration is not accurate enough, the sequence not long enough, or the motion sequence critical towards the set of constraints. More stable results are obtained when the principal point is considered as fixed in the centre of the image, but this assumption is not always fulfilled (see [19]) and is not accurate enough for image composition. Other attempts have been made to cope with varying internal parameters for AR applications [10]. However this approach uses targets arbitrarily positioned in the environment. It is therefore of limited use if outdoor scenes are considered.

In this paper we extend our previous works on vision based registration methods [12,13] to the case of zoom-lens cameras. Zoom-lens camera calibration is still found to be very difficult for several reasons [16,3]: modeling a zoom-lens camera is difficult due to optical and mechanical misalignments in the lens system of a camera. Moreover, zoom-lens variations can be confused with camera motions: for instance, it is difficult to discriminate a translation along the optical axis from a zoom.

In this paper, we take advantage of our application field to reduce the problem complexity. Indeed, we assume that the viewpoint and the focal length do not change at the same time. This assumption is compatible with the techniques used by professional movie-makers. We develop in this paper an original statistical approach: for each frame of the sequence, we test the hypothesis of a zoom against the hypothesis of a camera motion. If the motion hypothesis is retained, we still have to compute the camera pose with the old internal parameters. Otherwise, the internal parameters are computed assuming that the camera pose does not change. Camera parameters are supposed to be known in the first image of the sequence (they can be obtained easily from a set of at least 6 2D/3D point correspondences pointed out by the user).

This paper is organized as follows: first, we discuss in section 2 the pinhole camera model and we show the difficulties to recover both the camera pose and the internal parameters with varying focal lengths. Section 3 then describes our original method for zoom/motion partitioning of the sequence. Section 4 describes how registration is performed from this segmentation. Examples which demonstrate the effectiveness of our method are shown in section 5.

2 Registration Difficulties with a Zoom-Lens Camera

In this section, we first describe the pinhole model which is widely used for camera modeling. Then we describe our attempts to compute both the zoom and the motion parameters in a single stage. This task is called full calibration in the following. We show that classical registration methods fail to recover both the internal and the external parameters, even though some of the intrinsic parameters are fixed.

2.1 The Pinhole Camera Model

Let (X, Y, Z) represent the coordinates of any visible point M in a fixed reference system (world coordinate system) and let (X_c, Y_c, Z_c) represent the coordinates of the same point in the camera centered coordinate system. The relationship between the two coordinate systems is given by

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + T = [R\ T] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

where $[R, T]$ is the 3D displacement (rotation and translation) from the world coordinate system to the camera coordinate system.

We assume that the camera performs a perfect perspective transform with center O at a distance f of the image plane. The projection of M on the image plane is $(x = f\frac{X_c}{Z_c}, y = f\frac{Y_c}{Z_c})$. If $1/k_u$ (resp $1/k_v$) is the size of the pixel along the x axes (resp. y axes), its pixel coordinates are:

$$m = (k_u f \frac{X_c}{Z_c} + u_0, k_v f \frac{Y_c}{Z_c} + v_0) \tag{1}$$

where u_0, v_0 are the coordinates of the principal point of the camera (i.e. the intersection of the optical axis and the image plane).

The coordinates of a 3D point M in a world coordinate system and its pixel coordinates $m = \begin{pmatrix} u \\ v \end{pmatrix}$ are therefore related by $s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{A} [R\ T] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$

Full camera calibration amounts to compute 10 parameters: 6 external parameters (3 for the rotation and 3 for the translation) and 4 internal parameters ($\alpha_u = k_u f$, $\alpha_v = k_v f$, u_0 and v_0). Internal and external parameters are collectively referred to as camera parameters in the following.

2.2 Direct Full Calibration

When the internal parameters are computed off-line, the registration process amounts to compute the displacement $[R, T]$ which minimizes the re-projection error, that is the error between the projection of known 3D features in the scene and their corresponding 2D features detected in the image. For sake of clarity, we only suppose that the 3D features are points but we can also consider free form curves [12]. Moreover, we show in section 4 that 2D/2D correspondences can be added to improve the viewpoint computation.

The camera pose is therefore the displacement $[R, T]$ which minimizes the reprojection error

$$\min_{R,T} \sum dist(proj(M_i), m_i)^2$$

where minimization is performed only on the 6 external parameters (Euler angles and translation).

Theoretically, zoom-lens variations during shooting can be recovered in the same way. We have therefore to compute not only the camera viewpoint but also the internal camera parameters (focal length, pixel size, optical center) which minimize the reprojection error.

$$\min_{R,T,\alpha_u,\alpha_v,u_0,v_0} \sum dist(proj(M_i), m_i)^2$$

As mentioned by several authors [3], this approach is unable to recover both the internal and external parameters. To overcome this problem, some authors have proposed to reduce the number of unknowns by fixing some of the internal parameters to predefined values. As several experimental studies proved that the ratio $\frac{\alpha_u}{\alpha_v}$ remains almost constant during zoom variations [4], the set of the internal parameters to be estimated is then reduced to α_u, u_0, v_0. Unfortunately this approach fails to recover the right camera parameters. Consider for instance Fig. 1 which exhibits the results when registration is achieved on the 6 external parameters and the 3 internal parameters. As the house stands on a calibration target, the internal and external parameters can be computed for each frame using classical calibration techniques [6]. They can therefore be compared to those computed with the registration method. The camera motions with respect to the turntable and zoom variations during the *cottage sequence* are shown in Table 3.a. The camera trajectory along with the focal length computed for each frame of the sequence are shown in Fig. 1 in dashed lines. They have to be compared to the actual parameters which are shown in solid lines on the same figure. Note that the trajectory is the position of the camera in the horizontal plane and the arrows indicates the optical axis. These results prove that some camera motions are confused with zoom variations: besides the common confusion between zoom and translation along the optical axis, other motions do not correspond to the actual one: between the frames 13 and 14, an unexpected translation is detected and is compensated by a camera zoom out.

Such confusions are also observed in [3], but Bougnoux considers that they do not really affect the quality of the reconstruction of the scene. Unfortunately, the conclusion is not the same for the quality of a composition: an augmented sequence of the cottage using the computed viewpoints and focal length is shown on our web site. Small errors on the camera parameters do not really affect the reprojection of the scene but they induce jittering effects which affect the realism of the composition.

To take into account the interdependance of the internal parameters, Sturm expresses u_0 and v_0 as polynomial functions of α_u [16]. As the aspect ratio α_u/α_v remains constant over the sequence, only one internal parameter α_u has to be determined. However, to determine the degrees and the coefficients of the polynomial models, the camera has to be pre-calibrated for several zoom positions.

Hence, resolving the general full calibration problem is difficult. In this paper, we propose a robust solution to the particular case of sequences where camera

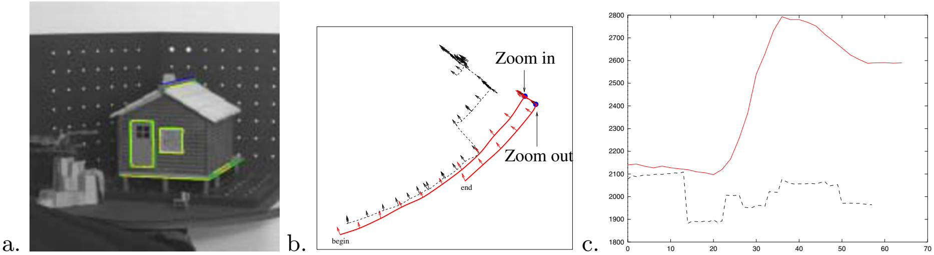


Fig. 1. (a) A snapshot of the cottage sequence and the reprojection of the 3D features. (b) The actual camera trajectory (solid line) and the computed one (dashed line). (c) The actual (solid line) and the estimated (dashed line) focal length during the sequence.

pose and zoom do not change at the same time. This particular case is very interesting for practical applications: indeed, when professional movie-makers make shootings, they generally avoid to mix camera motions and zoom variations. To take advantage of the structure of these sequences, we compute the reprojection error for each frame of the sequence in the two possible cases *zoom alone* and *camera motion alone*: (i) we consider that the internal parameters do not change and we search for the camera pose $[R, T]$ that minimizes the reprojection error (ii) we consider that the camera is fixed and we search for the internal parameters. Surprisingly, experiments we conducted show that the smallest of these two residuals does not always match the right camera parameters: Fig. 2 plots the reprojection error between frames 22 to 35 on a camera zoom sequence. For each frame i, the reprojection error between frame 20 and frame i is computed for the zoom and the motion hypothesis. This allows us to see the influence of the zoom magnitude on the criterion. The results prove that this method fails to recover the right camera parameters unless the magnitude of the zoom variation is high.

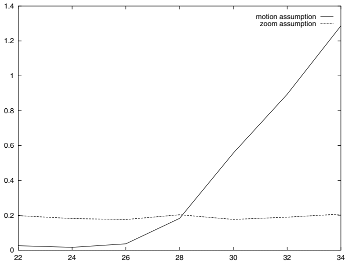


Fig. 2. Reprojection error with the zoom and the motion assumption for a camera zoom motion.

3 Discriminating between Zoom Variation and Camera Motion

The above results show that the classical registration methods cannot be used to cope with zoom-lens cameras. We therefore resort to a two-stage method: we first attempt to partition the video into camera motions and zoom variations. Then, our registration method is used on the image frames labeled *camera motion* while keeping the internal parameters constant, whereas the internal parameters are only computed for the frames labeled *zoom variations.* Unlike other methods for video partitioning which are based on the analysis of the optic flow [20], our method is only based on the analysis of a set of 2D corresponding points which are automatically extracted and matched between two consecutive images. The motion information brought by the key-point is very reliable and allows us to discriminate easily between zoom variation and translation along the optical axis. Our approach stands out from [20] in several points : in [20], the mean and the standard deviation of the optical flow are computed in seven non-overlapping sub-regions of the image. These values are compared with thresholds to discriminate between zoom, tilt, pan, Z-rotation, horizontal translation, vertical translation and Z-translation. However, it is not explained how the thresholds are computed, whereas it is the main point of the algorithm (furthermore, many confusions are observed in the final results). Moreover, to discriminate between a zoom and a Z-translation, the authors suppose that the center of the zoom is the center of the image, which is not true in practical situations [19].

Section 3.1 describes the way to extract key-points. Then we present the affine model of a zoom introduced in [4]. Finally we give our algorithm for zoom/motion automatic segmentation of the sequence (3.3).

3.1 Extracting and Matching Key-Points

Key-points (or interest points) are locations in the image where the signal changes two dimensionally: corners, T-junctions or locations where the texture varies significantly. We use the approach developed by Harris and Stephens [7]: they exploit the autocorrelation function of the image to compute a measure which indicates the presence of an interest point. More precisely, the eigenvalues of the matrix

$$\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \; (I_x = \frac{\partial I}{\partial x} \ldots)$$

are the principal curvatures of the auto-correlation function. If these values are high, a key-point is declared.

We still have to match these key-points between two consecutive images. To do this, we use correlation techniques as described in [21].

Fig 3.a and 3.b exhibit the key-points which have been automatically extracted in two successive images in the *loria scene* and Fig. 3.c shows the matched key-points.

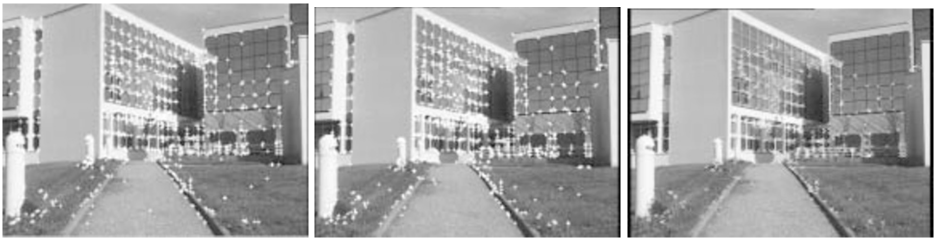


Fig. 3. (a,b) Key-points extracted in two consecutive frames. (c) The matched key-points.

3.2 Modeling Zoom-Lens Cameras

Previous studies on zoom-lens modeling proved that the ratio $\frac{\alpha_u}{\alpha_v}$ is very stable over long time periods. On the contrary, the position of the principal point (u_0, v_0) depends on the zooming position of the camera. This point can vary up to 100 pixels while zooming! However, for most camera lens, it can be shown that the principal point varies on a line while zooming [4]. That is the reason why an affine model with 3 parameters C_0, a_0, b_0 can be used to describe zoom variations. Enciso and Vieville [4] show that if (u', v') and (u, v) are corresponding points after zooming, we have

$$\begin{cases} u' = C_0 u + a_0, \\ v' = C_0 v + b_0. \end{cases} \tag{2}$$

The current matrix of the internal parameters A' is therefore deduced from the previous one A by:

$$A' = \begin{pmatrix} C_0 & 0 & a_0 \\ 0 & C_0 & b_0 \\ 0 & 0 & 1 \end{pmatrix} A. \tag{3}$$

If we want to use this property to discriminate between a zoom and a camera motion, we must prove that a camera motion can not be approximated by the same model. This can be shown from the equations of the optical flow : the optical flow (or instantaneous velocity) of an image point $(x = f\frac{X_c}{Z_c}, y = f\frac{Y_c}{Z_c})$, is

$$\begin{cases} \dot{x} = -\frac{U}{Z_c} + x\frac{W}{Z_c} + Axy - B(x^2+1) + Cy, \\ \dot{y} = -\frac{V}{Z_c} + y\frac{W}{Z_c} + A(y^2+1) - Bxy - Cx, \end{cases}$$

where $(U, V, W)^T$ is the translational component of the motion of the camera, $(A, B, C)^T$ is its angular velocity and f is set to 1 [8]. The optical flow obtained for the basic motions T_x (horizontal translation), T_y (vertical translation), T_z (Z-translation), R_x (tilt), R_y (pan) and R_z (Z-rotation) are given in table 1.a. Theoretically, none of these motions can be described by an affine transformation with three parameters. However, if $Z_c = Z_0 + \Delta Z$ where $\Delta Z \ll Z_0$ for each model point, that is the depth of the object is small with regard to the distance from the object to the camera (case 1), then T_x, T_y and T_z can be approximated by a zoom model whose parameters C_0, a_0 and b_0 are given in table 1.b (we use the approximation $\dot{x} = \frac{\dot{u}}{k_u} = \frac{u'-u}{k_u \Delta t}$ and $\dot{y} = \frac{v'-v}{k_v \Delta t}$). Moreover, if $x \ll 1$ and $y \ll 1$, that is the focal length is large (case 2), then R_x and R_y can also be approximated by a zoom model (see table 1.b).

Hence, some camera motions can induce an image motion close to the model of the zoom. Fortunately, most of them can easily be identified as camera motions. Indeed, for a zoom motion, the invariant point of the affine model $(\frac{a}{1-C_0}, \frac{b}{1-C_0})$ is the principal point of the camera and lies approximately in the middle of the image. On the contrary, for T_x, T_y, R_x and R_y, this point is outside the image and goes to infinity because C_0 is close to 1. Finally, only the translation along the optical axis T_z is really difficult to discriminate from a zoom.

a.

Motion	$\dot{x}$	$\dot{y}$
T_x	$-\frac{U}{Z_c}$	0
T_y	0	$-\frac{V}{Z_c}$
T_z	$x\frac{W}{Z_c}$	$y\frac{W}{Z_c}$
R_x	Axy	$A(y^2+1)$
R_y	$-B(x^2+1)$	$-Bxy$
R_z	Cy	$-Cx$

b.

Case	(C_0, a_0, b_0)
$T_x + case1$	$(1,-k_u\frac{U\Delta t}{Z_0},0)$
$T_y + case1$	$(1,0,-k_v\frac{V\Delta t}{Z_0})$
$T_z + case1$	$(1+\frac{W\Delta t}{Z_0},-u_0\frac{W\Delta t}{Z_0},-v_0\frac{W\Delta t}{Z_0})$
$R_x + case2$	$(1,0,A)$
$R_y + case2$	$(1,-B,0)$
-	-

Table 1. (a) Optical flow obtained for the basic motions. (b) Parameters of the approximating affine model for ambiguous cases.

3.3 Zoom/Motion Partioning

In this section, we present our approach for zoom/motion partioning. For each frame of the sequence, we test the hypothesis of a zoom against the hypothesis of a camera motion. We proceed as follows: key-points $(u_i, v_i)_{\{1 \le i \le N\}}$ and $(u'_i, v'_i)_{\{1 \le i \le N\}}$ are extracted and matched in two consecutive frames I_k and I_{k+1}. If we suppose that a zoom occurs, the model parameters C_0, a_0, b_0 which best fit the set of corresponding key-points are computed by minimizing the residual

$$r = \frac{1}{N}\sum_{i=1}^{N}(u'_i - C_0 u_i - a_0)^2 + (v'_i - C_0 v_i - b_0)^2. \tag{4}$$

We must now estimate the goodness of fit of the data to the affine model of the zoom. We have to test if the discrepancy r is compatible with the noise magnitude on the extracted key-points. Otherwise the zoom hypothesis should be questioned.

Statistical tests, such as χ^2 tests, are often used to estimate the compatibility of the data with the model with a given significance level a (90% for instance).

However, the standard deviation is needed for each datum. In our case, it is very difficult to calculate an error on the location of the key points. The χ^2 test has also a serious drawback: how can we set the significance level a? For a very large value of a, the hypothesis is always admitted, while for a very small value of a the hypothesis is always rejected.

That is the reason why we resort to another criterion to assess the zoom hypothesis. An important thing to note is that a zoom variation does not introduce new features in the images whereas translation motion does: some features which are visible for a camera viewpoint are no longer visible for a neighboring camera position. In Fig. 4.a, point A is not visible from C_k because it is occluded by the object O_1. But point A becomes visible when the camera moves from C_k to C_{k+1}. Note that such a phenomenon also arises for translation along the optical axis (Fig. 4.b). These features which become visible due to the camera motion are very important for assessing the zoom hypothesis. As key-points are not necessarily detected in the areas which become visible or which disappear, the key-points are not well suited for zoom assessment. We therefore use the set of all the contours detected in image I_k to assess the parameters (if $C_0 < 1$ we use image I_{k+1}). We first compute a correlation score for each contour. This score belongs to $[-1, 1]$ and is all the better that the zoom hypothesis is fulfilled. If the zoom hypothesis is satisfied, the gray levels $I_k(u, v)$ and $I_{k+1}(C_0 u + a_0, C_0 v + b_0)$ must be nearly the same. Moreover the neighborhood of these two corresponding points must be similar. We therefore use the correlation score to evaluate the zoom hypothesis. First, we define the correlation for a given point $m = (u, v)$ in I_k:

$$score(m) = \frac{\sum_{i,j=-n}^{i,j=n} I_k(u+i, v+j) \times I_{k+1}(C_0(u+i) + a_0, C_0(v+j) + b_0)}{(2n+1)^2 \sigma(I_k)\sigma(I_{k+1})},$$

where $\sigma(I_k)$ (resp. $\sigma(I_{k+1})$) is the standard deviation of I_k (resp. I_{k+1}) at point (u, v) in the neighborhood $(2n+1) \times (2n+1)$ of (u, v) (resp. $(C_0 u + a_0, C_0 v + b_0)$). The score ranges from -1 for two correlation windows which are not similar at all, to 1 for two correlation windows which are identical.

If a contour is given by the points $m_1, ..., m_p$, the score of a contour $\mathcal{C}$ is defined as the average of the scores of all points:

$$score(\mathcal{C}) = 1/p \sum_{i=1}^{i=p} score(m_i).$$

Finally the score of the *zoom hypothesis* is computed as the minimum of the score of each contour (note that only the strong contours are kept). This is a robust way to assess the zoom hypothesis. Indeed, if a zoom variation really happens, the score is high for each contour, and the global score is high too. On the contrary, if a camera motion happens, the score is generally low for nearly all the contours when the camera moves because the affine zoom model does not match the image transformation. Moreover, in case of a translating motion, the score is low for the contours of I_k which are occluded in I_{k+1}. Hence the global score is low too.

We still have to choose a threshold Th_{score} which allows us to distinguish between zoom variation and camera motion according to the global score. This

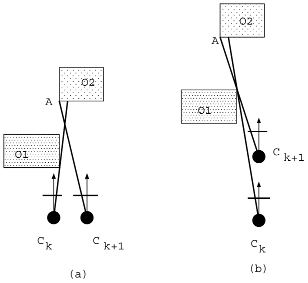


Fig. 4. New features appear under translating motion: point A is not visible from C_k but becomes visible from C_{k+1}.

value has been determined experimentally on various sequences. Experiments we have conducted (see section 5.2) prove that the value $Th_{score} = .5$ can be used for all the considered sequences to discriminate between zoom variation and camera motion even for the difficult case of a translation along the optical axis. Hence, if $global\_score > .5$ and if the invariant point of the affine model lies inside the image, then the zoom hypothesis is accepted, otherwise the camera motion hypothesis is retained.

4 Registration with a Zoom Lens Camera

Once the zoom/motion partitioning has been achieved, registration can be performed. If the frame belongs to a camera zoom sequence, then registration is performed only on the set of the internal parameters. Otherwise, registration is performed only on the set of the external parameters. As described in [12], we use n 2D/3D curve correspondences. Once the curves corresponding to the 3D features have been detected in the first frame of the sequence, they are tracked from frame to frame.

4.1 Registration for a Camera Motion

If the frame belongs to a camera motion sequence, we perform a six-parameters optimization from the curve correspondences:

$$\begin{array}{l} \alpha_u^{k+1} = \alpha_u^k, \alpha_v^{k+1} = \alpha_v^k, \\ u_0^{k+1} = u_0^k, v_0^{k+1} = v_0^k, \\ R^{k+1}, T^{k+1} = \underset{R,T}{argmin} \ \sum_i r_i^2, \end{array}$$

where r_i is a robust distance between 2D curve i and the projection of its 3-D counterpart. The computation of the residual r_i is detailed in [12]. However, one of the limitations of using 2D/3D correspondences originates in the spatial distribution of the model features: the reprojection error is likely to be large far from the 3D features used for the viewpoint computation. An example is shown in Fig 5.a: the viewpoint has been computed using the buiding in the background of the scene (the Opera). If we add a computer generated car on the foreground of the the scene, this car seems to hover.

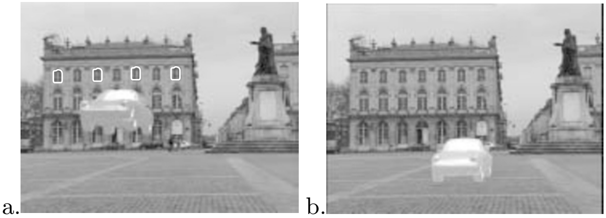


Fig. 5. (a) Registration using only 2D/3D correspondences. (b) Registration with the mixing method.

In order to improve viewpoint computation, we propose to use the key-points that have being matched for the partitionning stage. Previous approaches attempted to recover the viewpoint from 2D/2D correspondences alone [17]; unfortunately, this approach turns out to be very sensitive to noise in image measurements. For this reason, points correspondences between frames are here used to provide additional constraints on the viewpoint computation.

Our approach encompasses the strength of these two methods: the viewpoint is defined as the minimum of a cost function which incorporates 2D/3D correspondences between the image and the model as well as 2D/2D correspondences of key-points. Note that the extracted key-points bring information in areas where the 3D knowledge available on the scene are missing (fig. 5.b).

Given the viewpoint $[R_k, T_k]$ computed for a given frame k, we now explain how we compute the viewpoint in the next frame $k+1$ using the 3D model as well as the matched key-points $(q_k^i, q_{k+1}^i)_{1 \leq i \leq N}$. Let q_k^i be a point in frame k. Its corresponding point in frame $k+1$ belongs to the intersection of the image plane with the plane (C_k, C_{k+1}, q_k^i). This line is called the epipolar line. For two matched points (q_k^i, q_{k+1}^i), the quality of the viewpoint computed can be assessed by measuring the distance v_i between q_{k+1}^i and the epipolar line of q_k in frame k+1 [9]. Then, a simple way to improve the viewpoint computation using the interest points is to minimize

$$min_{R_{k+1}, T_{k+1}} \left(\frac{1}{n} \sum_{i=1}^{n} r_i^2 + \frac{\lambda}{N} \sum_{i=1}^{N} v_i^2 \right) . \qquad (5)$$

This way, any a priori information about the scene where the virtual object is going to sit on can be included in this model. The λ parameter controls the compromise between the closeness to the available 3D data and the quality of the 2D correspondences between the key-points. We use $\lambda = 1$ in our practical experiments. The minimum of equation 5 is computed by using an iterative algorithm for minimization such as Powell's algorithm, initialization being obtained from the parameters computed in the previous image of the sequence. More details about this method can be found in [13].

4.2 Registration for a Zoom

If the frame belongs to a camera zoom sequence, we get the new intrinsic parameters of the camera from equation 3. However, as approximation errors can propagate from frame to frame, we prefer to perform a three-parameters optimization from the 2D/3D correspondences. Hence, the camera parameters in frame $k+1$ are deduced from the camera parameters in frame k by the relation:

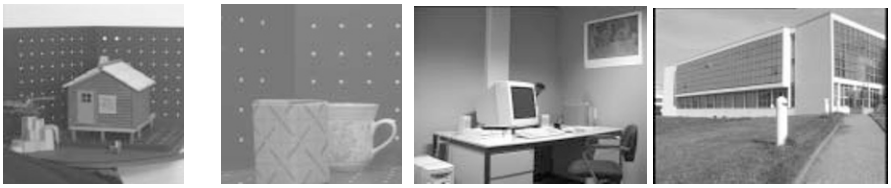
$$\begin{aligned} &R^{k+1} = R^k, T^{k+1} = T^k, \\ &C_0^{k+1}, u_0^{k+1}, v_0^{k+1} = \underset{C_0,u_0,v_0}{argmin}\ \textstyle\sum_i r_i^2, \\ &\alpha_u^{k+1} = C_0^{k+1}\alpha_u^k, \\ &\alpha_v^{k+1} = C_0^{k+1}\alpha_v^k. \end{aligned}$$

5 Experimental Results

In this section, we first justify experimentally the use of the threshold $Th_{score} =$ 0.5 to discriminate between zoom variations and camera motions. Then, section 5.2 present results of the partitioning process. Finally, registration results are given and augmented scenes are shown.

5.1 Choosing Th_{score}

To prove that $Th_{score} = 0.5$ is well suited to discriminate between camera motion and zoom variation, we considered a variety of video sequences (see Fig. 6). Each sequence alternates zoom variations with camera motions, including translations along the optical axis T_Z. For each frame of the sequence, the labeling in terms of *zoom variation, rotation motion, translation motion* is known. This allows us to compare the results of our algorithm with the actual ones.



1:The cottage sequence 2:The cup sequence 3:The office sequence 4:The Loria sequence

Fig. 6. Snapshots of the scenes used for testing the zoom/motion partitioning algorithm.

We first compute the score of the zoom hypothesis for each frame of the four sequences. Then we compute the mean along with the standard deviation of the score for the frames of the sequence corresponding to *zoom variation, rotation and translation* and (more difficult cases) *Z-translation* and *panoramic motion.* These results are shown in table 2: the first column shows the kind of variation undergone by the camera. The second and third columns give the scene under consideration and the number of frames in the sequence corresponding to the camera variation. Columns 4 and 5 show the mean and the standard deviation of the residual computed from the corresponding key-points (see equation 4). Finally, columns 6 and 7 shows the mean and the standard deviation of the score of the *zoom hypothesis.* These results clearly show that the use of the residual defined in equation (4) does not permit to discriminate between zoom variations and translation along the optical axis. On the contrary, the score we have defined gives high values when zoom happens and much smaller results when camera motion happens, even in case of T_Z translation. Finally, these experiments prove that the value $Th_{score} = .5$ is appropriate to distinguish zoom variations from camera motions.

variation in the camera parameters	scene	nb frames	r	σ_r	mean score	σ_{score}
Zoom	1	6	0.617	0.030	0.747	0.055
	2	4	0.460	0.266	0.860	0.055
	3	32	0.860	0.057	0.677	0.133
	4	29	0.515	0.014	0.561	0.064
Rotation + translation	1	10	3.593	1.439	-0.591	0.171
Translation along the optical axis	1	2	0.651	0.020	0.393	0.066
	2	4	0.841	0.018	0.274	0.035
	3	16	1.380	0.190	0.047	0.277
Panoramic motion	4	15	0.630	0.066	-0.209	0.315

Table 2. Score of the zoom hypothesis for various camera parameters.

5.2 Results in Zoom/Motion Partitioning

We now give detailed results of our algorithm on the *cottage sequence* and the *Loria sequence.* Note that the camera parameters are known for the *cottage sequence* because the house stands on a calibration target. The *Loria sequence* is a 700-frames sequence which has been shot outside our laboratory. The actual camera parameters are not available for this sequence, but we have manually partitioned the sequence (see table 3.b) to enable comparison with the algorithm.

For each of the two sequences (Fig. 7), we show the scores computed along the sequence, the results of our partitioning algorithm, and the computed zoom factor C_0. Also shown in the Fig. 7.b and 7.e is the actual partition of the sequence for comparison. For the *cottage sequence*, the algorithm performance is quite good and the computed parameters are very close to the actual parameters.

a.

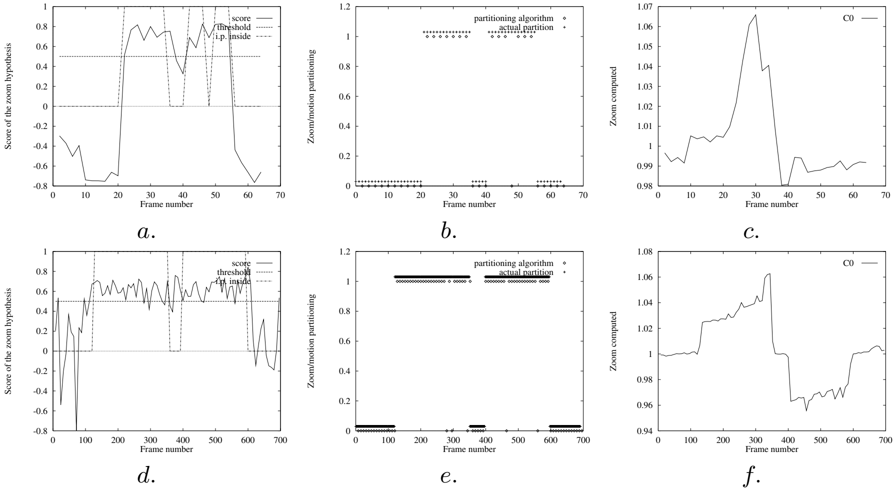
image	motion/zoom
0 → 20	rotation 40°
20 → 35	zoom in
35 → 40	translation 10cm
40 → 55	zoom out
55 → 65	rotation −20°

b.

Image frames	camera parameters
0 → 120	panoramic motion
121 → 344	Zoom in
345 → 408	no motion, nor zoom
409 → 600	Zoom out
601 → *end*	panoramic motion

Table 3. Camera parameters during (a) the cottage sequence and (b) the Loria sequence.

For the *Loria sequence*, the reader can notice that some scores are higher than the threshold during the panoramic motion between frames 0 and 100 (Fig. 7.d). However, in Fig. 7.a and 7.d, the test on the invariant point is shown with the dash-dot lines: the value 1 indicates that the invariant point is inside the image, while the value 0 indicates that the invariant point is outside the image. Using this constraint, the results of the partition process is very good (Fig. 7.b and 7.e).



a. *b.* *c.*

d. *e.* *f.*

Fig. 7. Results for the cottage sequence (first row) and the Loria sequence (second row).

5.3 Registration Results

In this section, registration results are shown for the *cottage sequence* and the *Loria sequence*. As the actual parameters are known for the *cottage sequence*, Fig.

8 shows the trajectory and the focal length computed with our algorithm (dashed lines) along with the actual parameters (solid lines). The reader can notice that the parameters obtained are in close agreement with the actual values. To prove the accuracy of the camera parameters, we have augmented the scene with a palm tree and a beach umbrella (Fig. 9). Note that the shadows between the scene and the computer generated objects greatly improve the realism of the composite images. They have been computed from a rough 3D reconstruction of the scene given by the corresponding key-points. The reprojection of the 3D model features with the computed camera parameters is also shown. The overall impression is very good.

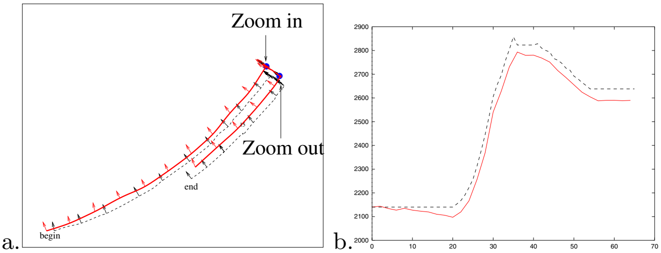


Fig. 8. Comparison of the actual trajectory (a) and focal length α_u (b) (solid lines) with the computed ones (dashed lines).

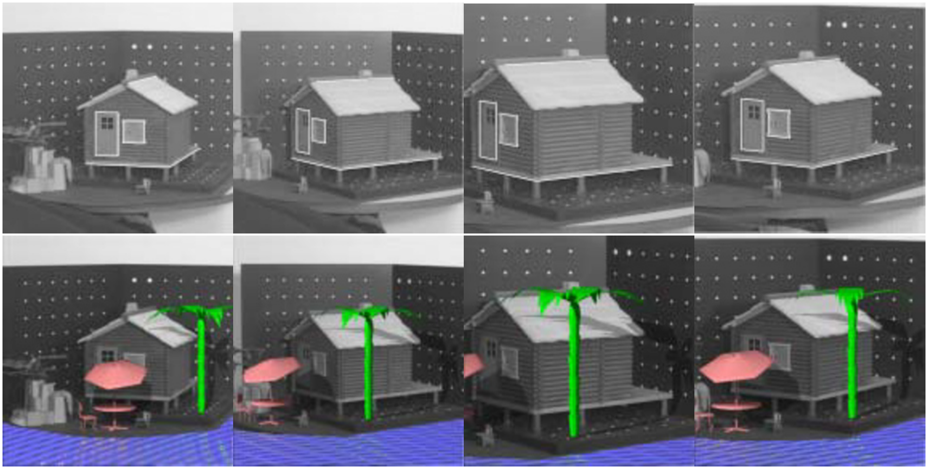


Fig. 9. Registration results on the cottage sequence: reprojection of the model (first row) and snapshots of the augmented scene (second row).

We do not have the actual camera parameters for the *Loria sequence*. Hence looking at the reprojection of the model features is a good way to assess the registration accuracy. Fig. 10 exhibits the reprojection of the model every hundred frames. The reader can notice that the reprojection error is small even at the end of the sequence, which proves the efficiency of our algorithm. Finally, we augment the sequence with the well known sculpture *La femme à la chevelure défaite* realized by *Mirõ*. The interested reader can look at the video sequences of our results at URL `http://www.loria.fr/~gsimon/eccv2000.html`.

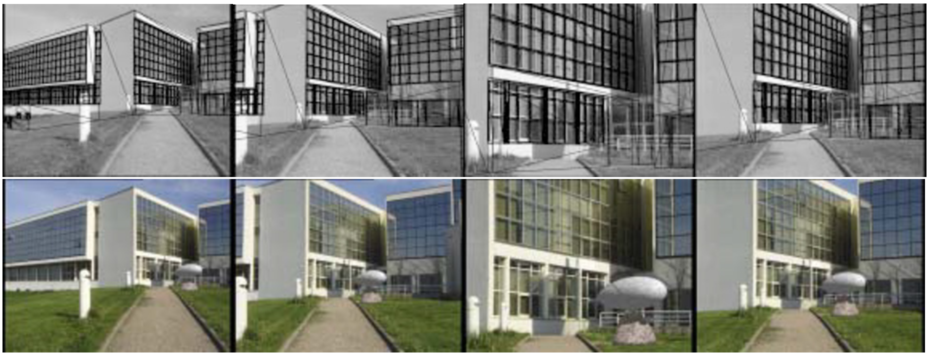


Fig. 10. Registration results on the Loria sequence: the reprojection of the model every hundred frames (first row) and snapshots of the augmented scene (second row).

6 Conclusion

In this paper we have presented an efficient registration algorithm for a zoom lens camera. We restricted our study to the case of image sequences which alternate zoom variation alone and camera motion alone. This is a quite reasonable assumption which is always fulfilled by professional movie-makers. The performance of our algorithm is quite good and our algorithm is capable of discriminating between zoom variations and T_Z translations. However, our experiments show that some improvements and extensions can be made to our approach.

First, experiments on the *Loria sequence* show that the camera trajectory is somewhat jagged. Smoothing the trajectory afterwards is not appropriate because the correspondences between the image and the 3D model are not maintained. We currently investigate methods to incorporate regularity constraints on the trajectory inside the registration process.

Second, as was observed in our experiments, moving objects in the scene may perturb the partitioning process. Indeed, the correlation score is always low for moving objects and this may lead to false rejection of the zoom hypothesis. Detecting moving objects in the scene prior to the registration process could help to solve this problem.

References

1. R. T. Azuma and G. Bishop. Improving static and dynamic registration in an optical see through display. In *Proc. SIGGRAPH'94*, pages 194–204. 578
2. M.-O. Berger, C. Chevrier, and G. Simon. Compositing Computer and Video Image Sequences: Robust Algorithms for the Reconstruction of the Camera Parameters. In *Proc. Eurographics'96*, volume 15, pages 23–32. 578
3. S. Bougnoux. From Projective to Euclidiean Space under any Practical Situation, a Criticism of Self-calibration. In *Proc. ICCV'98*, pages 790–796. 579, 581, 581
4. R. Enciso and T. Vieville. Self-calibration from four views with possibly varying intrinsic parameters. *Image and Vision Computing*, 15(4):293–305, 1997. 581, 583, 584, 584
5. G. Ertl, H. Müller-Seelich, and B. Tabatabai. MOVE-X: A System for Combining Video Films and Computer Animation. In *Proc. Eurographics'91*, pages 305–313. 578
6. O. D. Faugeras and G. Toscani. The Calibration Problem for Stereo. In *Proc. CVPR'86*, pages 15–20. 581
7. C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. Alvey Conference*, 1988. 583
8. B. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981. 584
9. Q. T. Luong. *Matrice fondamentale et calibration visuelle sur l'environnement, vers une plus grande autonomie des systèmes robotiques*. Thèse de doctorat, Université de Paris Sud, centre d'Orsay, December 1992. 588
10. J. Mendelsohn, K. Daniilidis, and R. Bajcsy. Constrained Self-Calibration for Augmented Reality Registration. In *Proc. IWAR'98*. 579
11. M. Pollefeys, R. Koch, and L. Van Gool. Self calibration and metric reconstruction in spite of varying and unknown camera parameters. In *Proc. ICCV'98*. 579
12. G. Simon and M.-O. Berger. A Two-stage Robust Statistical Method for Temporal Registration from Features of Various Type. In *Proc. ICCV'98*, pages 261–266. 578, 579, 580, 587, 587
13. G. Simon, V. Lepetit, and M.-O. Berger. Computer Vision Methods for Registration: Mixing 3D Knowledge and 2D Correspondences for Accurate Image Composition. In *Proc. IWAR'98*. 579, 589
14. A. State, G. Hirota, D. Chen, W. garett, and M. Livingston. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. In *Proc. SIGGRAPH'96*, pages 429–438. 578
15. A. State, M. Livingstone, W. Garett, G. Hirota, M. Whitton, and E. Pisan. Technologies for Augmented Reality Systems: Realizing Ultrasound Guided Needle Biopsies. In *Proc. SIGGRAPH'96*, pages 439–446. 578
16. Peter Sturm. Self Calibration of a moving Zoom Lens Camera by Pre-Calibration. In *Proc. BMVC'96*, pages 675–684. 579, 581
17. C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Method. *IJCV*, 9(2):137–154, 1992. 588
18. M. Uenohara and T. Kanade. Vision based object registration for real time image overlay. *Journal of Computers in Biology and Medecine*, 1996. 578, 578
19. R. G. Willson and S. A. Shafer. What is the Center of the Image? In *Proc. CVPR'93*, pages 670–671. 579, 583
20. W. Xiong and J.C.M. Lee. Efficient scene change detection and camera motion annotation for video classification. *Computer Vision and Image Understanding*, 71(2):166–181, 1998. 583, 583, 583
21. Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *Artificial Intelligence*, 78:87–119, 1995. 583