Robust Image Composition Algorithms for Augmented Reality

Marie-Odile Berger and Gilles Simon

INRIA Lorraine/ CRIN-CNRS, BP 101, 615 rue du Jardin Botanique, 54602 Vandœuvre les Nancy cedex, France

Abstract. We present our augmented reality system for image composition. We have worked with a view to avoiding strong and tedious interactions with the user. In this paper, we especially stress on the robust temporal registration method we have devised. An original method for resolving occlusions is also presented.

1 Introduction

In the past few years, virtual reality has attracted a great deal of media attention. The idea is to immerse a user into a completely computer-generated virtual world. Unfortunately, these environments often lack realism and the user is cut off from any view of the real world outside. Moreover, the numerical simulation of virtual environments is most of the time cost expensive.

On the contrary, augmented reality (AR) allows the user to interact with the real world in a natural way. Augmented reality systems aim at enhancing the user's vision with computer generated imagery but does not attempt to replace the real world. This explain why interest in AR has substantially increased in the past few years and medical, manufacturing or urban planning applications have been developed [11,5,3].

We focus in this paper on the problem of image composition for video sequences, which is one of the key point for numerous AR applications. Indeed, to make AR systems more effective, the computer generated objects must be blended convincingly with the real images.

Requirements for a realistic composition

The first challenge to be solved is to correctly retain up-to -date the scene-camera pose relationships over relative motion. This temporal registration allows the image of the computer generated objects to be computed for each frame of the video sequence. The registration task must be achieved with special care because the human visual system is very good at detecting even small misregistrations.

Unfortunately, ensuring temporal registration is not sufficient to perform realistic composition. Other significant visual cues to the human perceptual system must be considered: for instance, proper occlusion resolution between real and virtual objects is highly desirable in composition systems. Other photometric interactions between real and virtual objects (continuity of lighting, shadowing...) should also be considered.

Instrumenting the scene?
Augmented reality problems can often be solved by using either algorithmic solutions or sensor based solutions. For instance, the registration problem can be solved using position sensors (as Polhemus sensors). Easily detectable landmarks placed in the scene can also be used to make the registration process easier [5]. However, instrumenting the real world world is not always possible, especially for vast or outdoor environments. Thus, vision based object registration is an interesting and cheaper approach that leaves the environment unmodified. Hence, a wide variety of applications can be considered with such methods.

We focus in this paper on image composition methods which do not involve neither landmarks nor sensors. We only assume that the 3D model of some parts of the scene is known; it will be used for object based registration. This hypothesis is generally not restrictive for practical applications because the main structures of the scene are often known (ground, main objects in the scene ...).

We describe in the next section an overview of our augmented reality system. We then present the robust solutions we have devised for resolving the temporal registration problem (section 3) and the occlusion problem (section 4).

2 Overview of our Augmented Reality System

Before giving the overview of our system, we discuss the methods able to solve the temporal registration problem and the occlusion problem.

Camera calibration: If a large number of 2D/3D point or line correspondences are available, the registration reduces to a classical calibration process which allows the intrinsic parameters as well as the pose to be computed. Otherwise, a straightforward process is to calibrate the camera before shooting the video sequence. The underlying assumption is that the intrinsic parameters remain constant as the camera moves; we then have only to compute the camera pose for each frame. We use this latter solution because in practice, a small number of points can be extracted with sufficient accuracy, especially for outdoor scenes.

Viewpoint computation: Pose recovery has been extensively studied in the past few years. Two broad classes of methods can be distinguished: the most classical one uses object based registration; this means that 3D knowledge is needed to compute the pose from image/model correspondences. The other alternative is basically 2D: if the projection of a sufficient number of points are observed from different positions, the camera pose can be recovered up to a scale factor [10]. Unfortunately, these approaches turn out to be sensitive to inaccuracies in 2D feature measurements. For sake of efficiency, we therefore use object based registration.

Matching: Object based registration is a matching process between models and images. For video sequences, a reasonable assumption is that the user can locate objects in the first image frame. The matching process in the subsequent frames is then often achieved by using template matching (correlation). Since a single outlier can have a large effect on the resulting pose, special care

is often taken to reduce possible false matches. For instance [11] uses geometric invariants to check and select only successfully tracked points. Other methods [6] use a velocity model and a Kalman filter to better predict the position of the image feature. Unfortunately the use of a velocity model imposes regularity constraints on the camera motion; this can be inappropriate for augmented reality applications for which the scene is often shot by a moving observer.

We therefore advocate a less constraining approach. Instead of attempting to refine the matching process, we prefer to use a robust statistical method to compute the pose from the matching induced by the tracking process. Unlike most existing systems, registration is achieved from points, lines or free form curves. Another original aspect of our system lies in its ability to handle occlusions between the real scene and the computer generated objects.

The system is initialized with known camera parameters and a user specified set of four 3D-2D corresponding points pointed out by the user. This allows the initial pose to be computed (with the method of Dementhon and Davis). Then, the 2D features corresponding to the visible model features are automatically determined. Once initialized, the system follows a three step loop:

- **Tracking**: The set of features is tracked in the current image using a curve-based tracker that we have previously developed [2]. Among the set of tracked curves, a small number may be misdetected or completely erroneous (outlier). See for instance the primitives 4 and 5 in Fig. 2.d.
- **Robust temporal registration**: Correspondences are generally maintained during tracking. Unfortunately, even a single tracking error can have a large effect on the resulting pose. For point features, robust approaches allow the point to be categorized as outlier or not [7]. When curved features are considered, the problem is not so simple. We have then devised a robust algorithm capable of extracting the parts of the features that match the 3D model and to compute the pose in a robust manner (see section 3).
- **Resolving occlusion and image composition**: We propose in section 4 a contour based method that allows the occlusions to be solved without 3D reconstruction of the scene (see section 4).

3 Robust Statistical Methods for Temporal Registration

3.1 Robust Estimation

Pose recovery amounts to compute the rotation R and the translation t which map the world coordinate system on the camera coordinate system. $[R,t]$ is represented by 6 parameters $p = [p_1 ... p_6]$. For 2D/3D point correspondences, a classical way to compute the pose is to minimize the reprojection error

$$min_{\ p} \sum r_i^2 = min_{\ p} \sum d(m_i, proj(RM_i + t))$$

It is well known that the least square estimation is not robust to noise because the larger the residual r_i is, the larger is its influence on the estimate. To tackle

this issue, statisticians have suggested many robust estimators. Among them, the two most popular are the M estimator and the Least Median Square method (LMS) [8]. The LMS method consists in minimizing the median of the squared residuals $\min_{\mathbf{p}} \text{med}_i r_i^2$. This method is able to handle data sets which contain less than 50% outliers but is not very accurate. But the main drawback is that the minimum has to be searched in the space of possible estimates, that can be very large!

Since the rate of outliers is generally less than 50% for practical applications, we prefer to use M-estimators which can be reduced to a weighted least square problem. The M estimators try to reduce the effect of outliers by minimizing a function of the residuals

$$\min_{\mathbf{p}} \sum_{i=1}^{n} \rho(r_i), \tag{1}$$

where ρ is a continuous, symmetric function with minimum value at zero. Such estimators prove to be well suited to cases where the rate of outliers is approximately less than 20%. Table 1 lists three commonly used ρ functions and their derivative. Among these estimators, some are more restrictive than others: when Tukey's influence function is null for residuals larger than a threshold c, Cauchy's influence remains larger than zero while decreasing, whereas Huber's influence remains constant.

Table 1. Three commonly used M-estimators.

Type	$\rho(x)$	$\psi(x)$
Huber $\begin{cases} \text{if } \lvert x\rvert \le c \\ \text{if } \lvert x\rvert > c \end{cases}$	$\begin{cases} x^2/2 \\ c(\lvert x\rvert - c/2) \end{cases}$	$\begin{cases} x \\ c * \text{sgn}(x) \end{cases}$
Cauchy	$\frac{c^2}{2} \log\left(1 + \left(\frac{x}{c}\right)^2\right)$	$\frac{x}{1+\left(\frac{x}{c}\right)^2}$
Tukey $\begin{cases} \text{if } \lvert x\rvert \le c \\ \text{if } \lvert x\rvert > c \end{cases}$	$\begin{cases} \frac{c^2}{6}\left[1 - \left(1 - \left(\frac{x}{c}\right)^2\right)^3\right] \\ c^2/6 \end{cases}$	$\begin{cases} x\left(1 - \left(\frac{x}{c}\right)^2\right)^2 \\ 0 \end{cases}$

3.2 A Robust Two Stage Statistical Method for Pose Computation

In our system, features of various types are considered: point, lines and curves. Defining outliers for curved features is not so simple, as some parts of the 2D curves can perfectly match the 3D model whereas other parts can be erroneously matched. Let us define

- C_i be a 3D curve, described by the chain of 3D points $\{M_{i,j}\}_{1 \le j \le l_i}$
- c_i be the projection of C_i in the image plane, described by the chain of 2D points $\{m_{i,j}\}_{1 \le j \le l_i}$, where $m_{i,j} = Proj(\mathbf{R}M_{i,j} + \mathbf{t})$

– c'_i be the detected curve (tracked curve) corresponding to C_i, described by the chain of 2D points $\{m'_{i,j}\}_{1 \leq j \leq l'_i}$.

A simple solution would be to perform a one stage minimization

$$\min \sum_{i,j} \rho(d_{i,j}) \tag{2}$$

where $d_{i,j} = Dist(m'_{i,j}, c_i)$ is the distance between $m'_{i,j}$ and the curve c_i.

Unfortunately, this method is unsatisfactory because it merges all the features into a set of points, and makes no distinction between local errors (when a feature is only partially well localized), and gross errors (when the position of a feature is completely erroneous). However, these two kinds of errors are not identical, and not treating them separately induces a great loss of robustness and accuracy.

By contrast, we propose to perform a robust estimation in a two-stage process: a *local stage*, which computes a robust residual for each feature, and a *global stage* which minimizes a robust function of these residuals. The local stage reduces the influence of erroneous sections of the contours (features 1 and 4 on Figure 2.d), whereas the global stage discards the *feature outliers*, *i.e.* contours which are completely erroneous, or which contain too large a portion of erroneous points (feature 5 on Figure 2.d).

The local stage

The aim of this stage is to reduce the influence of erroneous sections of the features: to perform this task, the residual error r_i of curve C_i is computed by a robust function of the distances $\{d_{i,j}\}_{1 \leq j \leq l'_i}$. We use the M-estimation technique by taking $r_i^2 = \frac{1}{l'_i} \sum_{j=1}^{l'_i} \rho(d_{i,j})$. Since this estimate must not be too restrictive, we have hence chosen Huber's function for the local stage , which has proved to be a good choice in our experiments [9].

The global stage

This stage fits the tracked 2D features to the projection of the 3D features, by minimizing the residuals r_i which are computed for each couple of 3D/2D features.We use Tukey's function, which is restrictive enough to suppress the influence of outliers, but which takes all the data into consideration.

3.3 Results

We present in this section an application of our method to an augmented reality application: the illumination of the bridges of Paris [3]. The aim was to test several candidate illumination projects for a number of bridges of the Seine. We therefore want to replace the bridge in the sequence with its lighting simulation. A 300-image panoramic sequence of the Pont Neuf was shot at dusk time from another bridge. Because the images are dark and noisy, only 6-8 curves can be tracked in each frame (Figure 2.b). The solid lines correspond to the tracked 2D features, whereas the dashed lines correspond to the projection of the corresponding model features (black is used for the features which are not - yet - used). The result of the tracking in the 12^{th} image is shown in Figure 2.c. The

reader may notice that the tracking process fails for feature 5. Figure 2.d shows the re-projection of the model features after the robust pose computation. Despite the bad accuracy of the model, the result is visually convincing. In order the reader to be aware of the parts of the curve which are less taken into account in the computation, we have drawn in black the points for which the residual is greater than c (c is defined in Table 1. Roughly speaking, these points are the ones for which the weight in the computation is decreased because their residual is too large. It must also be noticed that feature 5 is considered as an outlier and is removed from the set of tracked features (discarded features are drawn in black).

Since new features may appear while old ones disappear, the set of model features that are tracked in the sequence must be dynamically updated. The method we use to achieve this task is described in [9].

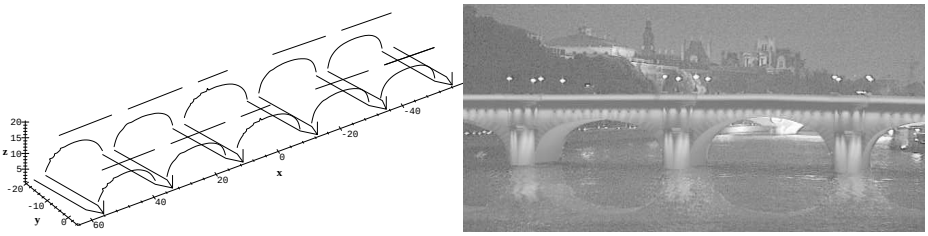

Fig. 1. (a)The complete wire-frame model of the bridge; (b)Final composition.

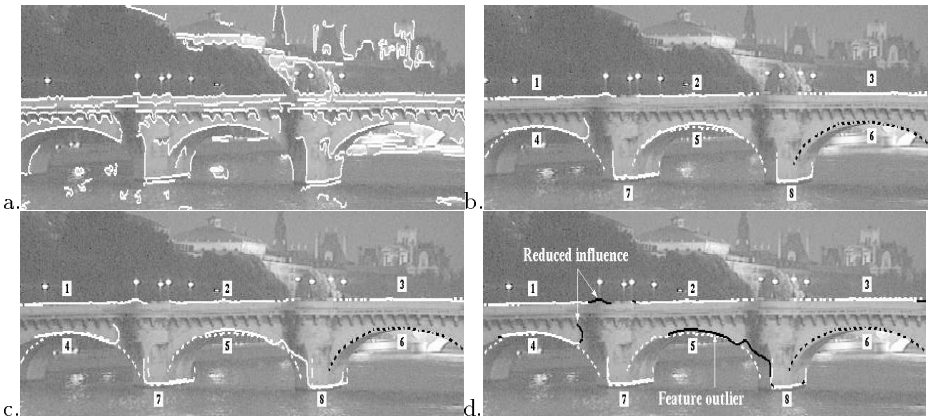

Fig. 2. Temporal registration for the 12^{th} image. (a) Edge map. (b) 2D features before tracking (image 10). (c) Tracking in image 12 (projections of the 3D features are those of image 10). (d) Robust pose computation.

4 Resolving Occlusions

Most of the time, augmented reality systems simply overlay computer generated images and only attempt to minimize object registration errors. However, such methods are effective only when there are no occlusions between the real and the computer generated objects.If the model of the 3D scene is known, as in [4], the problem can easily be solved. Otherwise, resolving occlusion could theoretically be achieved by inferring a dense map from two consecutive images. Unfortunately, despite new advances in 3D reconstruction, the depth map lacks accuracy and cannot be used as is. Instead of performing 3D reconstruction, we propose to use a contour based approach. Our aim is to find, among the contours in the scene, those belonging to the boundary of the occlusion mask. Our approach stems from the fact that for a real scene containing only rigid objects, the boundary of the mask is only composed of contours present in the image and of occluding contours of the virtual object. This is obviously not true anymore if the objects are deformable or can penetrate each other [1].

The main steps of our algorithm are summarized below [1].

The contour chains are extracted in the region to which the computer generated object V corresponds. These contours are tracked in the next image. Finally, the matching of the contours points between the two images is performed by using the epipolar constraint. Two corresponding points are denoted by (m_1, m_2) in the sequel.

The heart of our system is the labeling stage which allows each contour point m_1 to be labeled with *in front of* or *behind* depending on the relative position of the corresponding point of the scene and of the computer generated object. To this aim, let us define f_{m_1} (Fig. 3):

$$f_{m_1} : Z \rightarrow proj_{I_2}(m1_x, m1_y, Z)$$

where $m_1 = (m1_x, m1_y)$ and $proj_{I_2}$ is the projection in image I_2. It can easily be seen that $f_{m_1}(Z)$ can be expressed as an homographic function of Z whose coefficients depend on the calibration process and on the image point m_1.We have

$$m_2 = f_{m_1}(Z_{real}) \quad and \quad m_{obj} = f_{m_1}(Z_{obj})$$

Due to the monotony of the homography, it is therefore easy to compare Z_{real} and Z_{obj}.

The last step is to recover the occluding mask from the set $\mathcal{H}$ of contour points labeled *in front of*. Because some labeling errors may occur, we resort to a regularization approach. The underlying idea is to add regularity constraints which will produce the most regular curve resting on $\mathcal{H}$. Starting from a closed curve outside $\mathcal{H}$, we use active contour models to obtain such a result. An example is shown in Fig 3.b.

5 Conclusion

This paper has presented an image composition system capable of ensuring temporal registration in a robust and autonomous way. Significant results on video

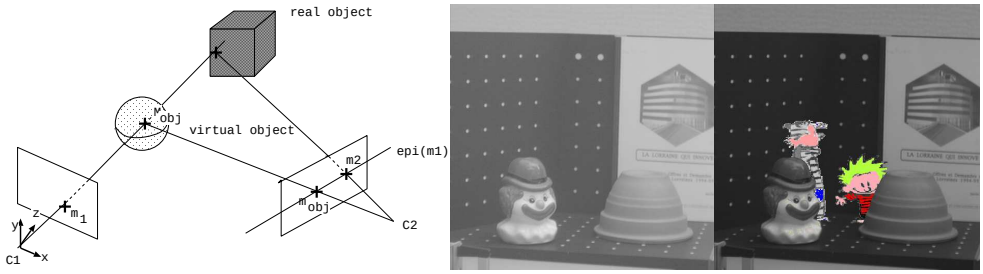

Fig. 3. (a)The relative positions of the real and the virtual objects; (b-c) An example of image composition.

image sequences can be seen at URL http://www.loria.fr/isa. For architectural applications, the size of the database which describes the objects is sometimes huge. Thus, further investigations concern the way to infer the more pertinent 3D features to be tracked.

References

1. M.-O. Berger. Resolving occlusion in augmented reality : a contour based approach without 3D reconstruction. In *CVPR 97, Puerto Rico (USA)*, pages 91–96, 1997.
2. M.-O. Berger. How to Track Efficiently Piecewise Curved Contours with a View to Reconstructing 3D Objects. In *ICPR 94, Jerusalem (Israel)*, volume 1, pages 32–36, 1994.
3. M.-O. Berger, C. Chevrier, and G. Simon. Compositing Computer and Video Image Sequences: Robust Algorithms for the Reconstruction of the Camera Parameters. In *Eurographics'96, Poitiers, France*, volume 15, pages 23–32, August 1996.
4. D. Breen, R. Whitaker, E. Rose, and M. Tuceryan. Interactive Occlusion and Automatic Object Placement for Augmented Reality. In *Eurographics'96, Poitiers, France*, 1996.
5. G. Ertl, H. Müller-Seelich, and B. Tabatabai. MOVE-X: A System for Combining Video Films and Computer Animation. In *Eurographics*, pages 305–313, 1991.
6. D. Koller, K. Daniilidis, and H. H. Nagel. Model-Based Object Tracking in Traffic Scenes. In *ECCV 92, Santa Margherita Ligure (Italy)*, pages 437–452, 1992.
7. S. Ravela, B. Draper, J. Lim, and R. Weiss. Tracking Object Motion Across Aspect Changes for Augmented Reality . In *ARPA Image Undertanding Worshop, Palm Spring (USA)*, August 1996.
8. P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley Series in Probability and Mathematical Statistics. Wiley, 1987.
9. G. Simon and M.-O. Berger. A two-stage robust statistical method for temporal registration from features of various type. In *ICCV 98, Bombay (India)*, 1998.
10. C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Method. *IJCV*, 9(2):137–154, 1992.
11. M. Uenohara and T. Kanade. Vision based object registration for real time image overlay. *Journal of Computers in Biology and Medecine*, 1996.