Registration methods for harmonious integration of real worlds and computer generated objets

G. Simon, V. Lepetit and M.-O. Berger

LORIA, BP 239,
54506 Vandoeuvre-les-Nancy, France
e-mail: {gsimon, lepetit, berger}@loria.fr

Abstract

In this paper, we present vision based methods appropriate for image composition. We especially address the registration process with a zoom lens camera. We also describe how to solve possible occlusions between the scene and the computer generated objects.

Augmented reality systems aim at enhancing the user's vision with computer generated imagery but do not attempt to replace the real world. 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. We focus in this paper on the problem of adding computer-generated objects (also called virtual objects) in video sequences. This is one of the key-points for numerous AR applications. 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. Realistic merging of virtual and real objects also requires that objects behave in a physical plausible manner in the environment: they can be occluded by objects in the scene, they are shadowed by other objects ...

As we are interested in general application settings, we have developed automatic vision based methods appropriate for image composition. Note that sensor based solutions have also been developed but they cannot be used for vast or outdoor environments. In this paper we mainly focus on the registration problem because it is one of the most basic challenge in augmented reality. We propose a robust method for viewpoint computation which utilizes 3D knowledge on the scene as well as 2D/2D correspondences of key-points that are automatically extracted and matched between two consecutive frames. We recently extended this method to the case of an unknown focal length varying from image to image. Finally, we briefly describe how to solve possible occlusions between the computer generated objects and the real scene. Various results and videos can be viewed at URL `http://www.loria.fr/~gsimon/eg99.html`.

1. Robust pose computation

The 3D model of some objects in the scene to be augmented is most of the time available. These 3D data can be used to compute the viewpoint provided that their corresponding 3D features can be identified in the images. We recently proposed in [4] a robust registration method which allows us to compute the viewpoint from 2D/3D correspondences of various features: points, lines and free form curves. Our method minimizes the reprojection error of the model features in the image. However, one of the limitations of this method originates in the spatial distribution of the model features: the reprojection error is likely to be large for the 3D features far from those used for the viewpoint computation. An example is shown in Fig 2.a: the viewpoint has been computed using the four curves on the building in the background of the scene (the Opera). We add a computer generated car on the square which moves from the background to the foreground of the scene. As the car moves away from the Opera, the reprojection error increases and the car seems to hover.

In order to improve viewpoint computation, we propose in this paper to use 2D/2D point correspondences between consecutive frames. Previous works attempted to recover the viewpoint from 2D/2D correspondences alone [5]; unfortu-

Published by the Eurographics Association, ISSN 1017-4656.

nately, 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 exploits 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 that are automatically extracted and matched in two consecutive frames. Note that the extracted key-points bring information in areas where the 3D knowledge available on the scene are missing.

1.1. Extracting and matching key-points

Key-points (or interest points) are locations in the image where the signal changes two dimensionally: corner, T-junctions, locations where the texture varies significantly... We use the approach developed by Harris and Stephens [3]: they use 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

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

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

1.2. Mixing 3D knowledge and points correspondences

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 $(q_k^i, q_{k+1}^i)_{1 \leq i \leq m}$. Before describing the cost function to be minimized, we first recall in Fig. 1 the relationships between two matched key-points q_k, q_{k+1} corresponding to the 3D point Q and the two viewpoints $[R_k, t_k]$ and $[R_{k+1}, t_{k+1}]$. Let q_k be a point in I_k; its corresponding point in I_{k+1} belongs to the intersection of the image plane with the plane (C_k, C_{k+1}, q_k). This line is called the epipolar line (Fig. 1.a) . For two matched points (q_k, q_{k+1}), the quality of the viewpoint computed can be assessed by measuring the distance v between q_{k+1} and the epipolar line of q_k in I_{k+1} (Fig. 1.b).

Then, a simple way to improve the viewpoint computation using the interest points is to minimize

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

where

- r_i is the distance in frame $k+1$ between the image features and the projection of the model features (Fig. 1.c),
- v_i measures the quality of the computed position of the camera
- In practice, we often use $\rho(r_i)$ and $\rho(v_i)$ instead of the squared residuals, where ρ is a robust statistical estimator (M-estimator). This way, possible matching errors have little influence on the result.

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 interest of the mixing algorithm is shown in Fig. 2. The car and the scene are combined seamlessly and the realism of the composition is very good.

2. Registration with a varying focal length

We now extend our approach to the case of a camera with a varying focal length. We have therefore to compute not only the camera viewpoint but also the intrinsic camera parameters (focal length, size of the pixel, optical center). In this paper, 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.

Previous studies on zoom-lens cameras [1] prove that the image transformation resulting from varying focal length can be described using an affine model with 3 parameters C_0, a_0, b_0: if (u', v') and (u, v) are corresponding points after zooming, we have

$$\begin{aligned} u' &= C_0 u + a_0 \\ v' &= C_0 v + b_0 \end{aligned}$$

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) and (u_i', v_i') are extracted and matched in two consecutive frames I_k and I_{k+1}. A least squares estimation allows us to compute the model parameters C_0, a_0, b_0 which best fits the set of corresponding key-points. We must now estimate the goodness of fit of the data to the affine model of the zoom. To do this, we evaluate the zoom hypothesis on the set of contours detected in the image. If the correlation

$$\sum_{(u,v)\ contour\ points} |I_{k+1}(C_0 u + a_0, C_0 v + b_0) - I_k(u, v)|$$

is sufficient, the hypothesis of a zoom is accepted, and the intrinsic parameters are updated accordingly (the new focal length f' is deduced from the old one f with the relation $f' = C_0 \times f$). Otherwise, we consider that the camera moves and the camera viewpoint is computed using the algorithm described in section (1). Significant results of our algorithm are shown on the *cottage sequence* at our URL. These results clearly prove that our algorithm is able to discriminate between focal changes and camera motions even in the difficult case where the camera translates along the optical axis.

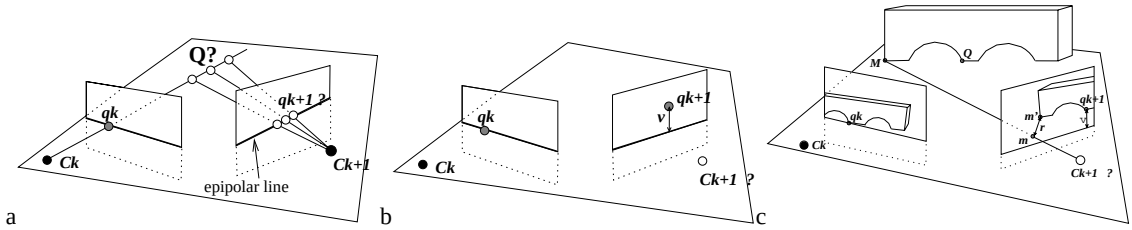

Figure 1: *(a,b)Constraints between matched points; (c) the residuals used in the mixing method.*

Figure 2: *(a) Registration using only 2D/3D correspondences; (b) Registration with the mixing method (c) Occlusion resolution.*

3. Resolving occlusions

Once the viewpoint has been computed for each frame of the sequence, the computer generated objects can be added in the scene. However, as the virtual object can be occluded by real objects in the scene, we have to determine the visible part of the virtual object.

To do this, 3D stereo reconstruction of the scene is performed in the area where the virtual object must be added. Ideally, depth should be computed at every point of this area. However, for sake of accuracy, we prefer to compute the depth only for features which are well matched. Therefore, depth is computed for contour points and key-points. For each feature point, the estimated depth is then compared with the depth of the virtual object. This allows us to obtain a set of contours and key-points $\mathcal{O}$ which stand in front of the computer generated object. We still have to determine the shape of the occluding object from this set of points. This amounts to produce the border of $\mathcal{O}$ reasonably close to the one perceived by human visual system. Using a slightly modified version of the algorithm described in [2], we obtain quite satisfying occlusion masks: Fig. 2.c exhibits the set $\mathcal{O}$ and the occlusion mask when adding a virtual helicopter in the scene.

4. Conclusion

We have presented several techniques for harmonious integration of computer generated objects on video sequences. These methods make the composition task easier. Indeed, the registration task is fully automatic and is able to handle focal length variations during shooting. Currently, the time needed to process one frame is around 3 s in the sequence considered in Fig. 2. In any case, our methods can be very useful to perform post production tasks: visual assessment of new projects in their final settings, special effects in movies...

References

1. R. Enciso and T. Vieville. Self-calibration from four views with possibly varying intrinsic parameters. *Image and Vision Computing*, 15(4):293–305, 1997.

2. G. Garai and B. Chaudhuri. A Split and Merge Procedure for Polygonal Detection of Dot Pattern. *Image and Vision Computing*, 17:75–82, 1999.

3. C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proceedings of 4th Alvey Conference*, Cambridge, August 1988.

4. 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)*, pages 261–266, January 1998.

5. C. Tomasi and T. Kanade. Shape and Motion from Image Streams under Orthography: A Factorization Method. *International Journal of Computer Vision*, 9(2):137–154, 1992.