Augmented Reality using Uncalibrated Video Sequences

Kurt Cornelis*, Marc Pollefeys*, Maarten Vergauwen and Luc Van Gool

K. U. Leuven, ESAT-PSI
Kardinaal Mercierlaan 94, B-3000 Leuven, Belgium
kcorneli | pollefey | vergauwe | vangool@esat.kuleuven.ac.be
WWW home page:http://www.esat.kuleuven.ac.be/~kcorneli

Abstract. *Augmented Reality*(AR) aims at merging the real and the virtual in order to enrich a real environment with virtual information. Augmentations range from simple text annotations accompanying real objects to virtual mimics of real-life objects inserted into a real environment. In the latter case the ultimate goal is to make it impossible to differentiate between real and virtual objects. Several problems need to be overcome before realizing this goal. Amongst them are the rigid registration of virtual objects into the real environment, the problem of mutual occlusion of real and virtual objects and the extraction of the illumination distribution of the real environment in order to render the virtual objects with this illumination model. This paper will unfold how we proceeded to implement an *Augmented Reality System* that registers virtual objects into a totally uncalibrated video sequence of a real environment that may contain some moving parts. The other problems of occlusion and illumination will not be discussed in this paper but are left as future research topics.

1 Introduction

1.1 Previous Work

Accurate registration of virtual objects into a real environment is an outspoken problem in Augmented Reality(AR). This problem needs to be solved regardless of the complexity of the virtual objects one wishes to enhance the real environment with. Both simple text annotations and complex virtual mimics of real-life objects need to be placed rigidly into the real environment. *Augmented Reality Systems* that lack this requirement will demonstrate serious 'jittering' of virtual objects in the real environment and will therefore fail to give the user a real-life impression of the augmented outcome.

The registration problem has already been tackled by several researchers in the AR-domain. A general discussion of all coordinate frames that need to be registered with each other can be found in [25]. Some researchers use predefined geometric models of real objects in the environment to obtain vision-based object registration [15, 22, 27]. However, this delimits the application of such systems because geometric models of real objects in a general scene are not always readily available. Other techniques

* Kurt Cornelis and Marc Pollefeys are respectively research assistant and postdoctoral fellow of the Fund for Scientific Research - Flanders(Belgium)(F.W.O. - Vlaanderen)

have been devised to make the calibration of the video camera obsolete by using affine object representations [16]. These techniques are simple and fast but fail to provide a real impression when projective skew is dominant in the video images. Therefore virtual objects can be viewed correctly only from large distances where the affine projection model is almost valid. So it seems that the most flexible registration solutions are those that don't depend on any a priori knowledge of the real environment and use the full perspective projection model. Our AR-System belongs to this class of flexible solutions.

To further enhance the real-life impression of an augmentation the occlusion and illumination problems need to be solved. The solutions to the occlusion problem are versatile. They differ in whether a 3D reconstruction of the real environment is needed or not [3, 5]. Also the illumination problem has been handled in different ways. A first method uses an image of a reflective object at the place of insertion of the virtual object to get an idea of the incoming light at that point [6]. A second approach obtains the total reconstruction of a 3D radiance distribution by the same methods used to reconstruct a 3D scene [19]. Another approach consists of the approximation of the illumination distribution by a sphere of illumination directions at infinity [20].

As Computer Generated Graphics of virtual objects are mostly created with non physically-based rendering methods, techniques that use image-based rendering can be applied to incorporate real objects into another real environment [23] to obtain realistic results. Image-based rendering is explained in [7].

However, the 'jittering' of virtual objects in the real environment can degrade severely the final augmented result, even if problems of occlusion and illumination can be resolved exactly. We focussed on developing an AR-System that solves the registration problem as a prerequisite. It is based primarily on a 3D reconstruction scheme that extracts motion and structure from uncalibrated video images and uses the results to incorporate virtual objects into the real environment.

1.2 Overview

In the first upcoming section we will describe the motion and structure recovery algorithm of the AR-System. Although the main goal is the recovery of motion of the camera throughout the video sequence, the system also recovers a crude 3D structure of the real environment. This can be useful to handle future problems like resolving occlusions and extracting the illumination distribution of the real environment. We will focus on the motion recovery abilities of the AR-System.

In a following section we will discuss the use of the recovered motion parameters and the 3D structure to register virtual objects within the real environment. This involves using the crude 3D representation of the real environment which we obtain as an extra from the motion recovery algorithm. Dense 3D reconstruction of the real environment is not necessary but may prove useful for future solutions to the occlusion problem.

Another section will give an overview of the final AR-algorithm. We will finish by showing results of the AR-System on some applications and by indicating future work to be done in order to upgrade the AR-System.

2 Motion and Structure Recovery

2.1 Preliminaries

As input to the AR-System we can take totally uncalibrated video sequences. The video sequences are neither preprocessed nor set up to contain calibration frames or fiducial markers in order to simplify motion and structure recovery. Extra knowledge on calibration parameters of the video camera can be used to help the AR-System to recover motion and structure but is not necessary to obtain good results.

The video sequences are not required to be taken from a purely static environment. As long as the moving parts in the real environment are small in the video sequence the algorithm will still be able to recover motion and structure.

2.2 Motion and Structure Recovery Algorithm

Image Features Selection and Matching Recovery of motion in Computer Vision is almost always based on tracking of features throughout images and uses these to determine motion parameters of the camera viewing the real environment. Features come in all flavours like points, lines, curves [4] or regions [26]. The features we use are the result of the Harris Corner Detector algorithm [9] applied to each image of our input video sequence. The result consists of points or *corners* in the images determining where the image intensity changes significantly in two orthogonal directions.

We end up with *corners* in each image of the video sequence but these are still unmatched from one image to another. We need to match them in different images in order to extract motion information. An initial set of possible matching corners is constructed using a small search region around each corner looking for corners in other images which have a large normalized intensity cross-correlation with the corner under scrutiny. Corresponding or *matching* corners are constrained through epipolar geometry to lie on each others epipolar line. This constraint can be expressed in terms of a linear equation between the two images one wishes to match the corners from:

$$x_1{}^T \mathbf{F}_{12} x_2 = 0 \tag{1}$$

where $x_1 = (u_1, v_1, 1)^T$ and $x_2 = (u_2, v_2, 1)^T$ denote homogeneous image coordinates of matching corners in the first and second image. $\mathbf{F}_{12}$ is a 3×3 singular matrix which describes the epipolar geometry between the two images. The epipolar line from corner x_1 in image 2 and from corner x_2 in image 1 can be written down respectively as:

$$\mathbf{F}_{12}^T x_1 = 0 \text{ and} \tag{2}$$

$$\mathbf{F}_{12} x_2 = 0 \tag{3}$$

Using equation (1) each possible match between corners from the two images adds a constraint on the elements of the matrix $\mathbf{F}_{12}$. Extra constraints can be superimposed on $\mathbf{F}_{12}$ due to its singular nature and because it can only be determined up to a scalefactor as we are working with homogeneous image coordinates. Several algorithms have

been devised to determine reliable matches between the corners of two images. These matches lead to a reasonable consistent $\mathbf{F}_{12}$, which means that equation (1) returns a small residual error for an important fraction of the presumed matches. The determination of this particular set of matches is achieved by a RANSAC algorithm [12] which determines $\mathbf{F}_{12}$ from trial matches and additional constraints of singularity and scalability. Once a good initial $\mathbf{F}_{12}$ is obtained it is optimized using all consistent matches and a Levenberg-Marquardt optimization technique.

As long as the moving parts in the real environment are small in the video sequence the RANSAC algorithm will treat corners belonging to these moving parts as outliers. They will be properly discarded in the determination of the matrix $\mathbf{F}_{12}$ and the matching corners.

Initializing Motion and Structure Recovery Once corner matches between two initial images are found, they can be used to initialize motion and structure recovery from the video sequence.

The relation between a 3D structure point and its projection onto an image can be described by a linear relationship in homogeneous coordinates:

$$m_k \sim \mathbf{P}_k M \tag{4}$$

in which $M = (X, Y, Z, 1)$ and $m_k = (x_k, y_k, 1)^T$ are the homogeneous coordinates of the 3D structure point and its projection onto image k respectively. $\mathbf{P}_k$ is a 3×4 matrix which describes the projection operation and '$\sim$' denotes that this equality is valid up to a scalefactor.

The two initial images of the sequence are used to determine a reference frame. The world frame is aligned with the camera of the first image. The second camera is chosen so that the epipolar geometry corresponds to the retrieved $\mathbf{F}_{12}$.

$$\begin{array}{l} \mathbf{P}_1 = [\ \mathbf{I}_{3\times 3} \mid 0_3\ ] \\ \mathbf{P}_2 = [\ [\mathsf{e}_{12}]_\times \mathbf{F}_{12} + \mathsf{e}_{12}\pi^\top \mid \sigma \mathsf{e}_{12}\ ] \end{array} \tag{5}$$

where $[\mathsf{e}_{12}]_\times$ indicates the vector product with e_{12}. Equation (5) is not completely determined by the epipolar geometry (i.e. $\mathbf{F}_{12}$ and e_{12}), but has 4 more degrees of freedom (i.e. π and σ). π determines the position of the reference plane (this corresponds to the plane at infinity in an affine or metric frame) and σ determines the global scale of the reconstruction. To avoid some problems during the reconstruction it is recommended to determine π in such a way that the reference plane does not cross the scene. Our implementation uses an approach similar to the quasi-Euclidean approach proposed in [2], but the focal length is chosen so that most of the points are reconstructed in front of the cameras[1]. This approach was inspired by Hartley's cheirality [10]. Since there is

[1] The quasi-Euclidean approach computes the plane at infinity based on an approximate calibration. Although this can be assumed for most intrinsic parameters, this is not the case for the focal length. Several values of the focal length are tried out and for each of them the algorithm computes the ratio of reconstructed points that are in front of the camera. If the computed plane at infinity –based on a wrong estimate of the focal length– passes through the object, then many points will end up behind the cameras. This procedure allows us to obtain a rough estimate of the focal length for the initial views.

no way to determine the global scale from the images, σ can arbitrarily be chosen to $\sigma = 1$.

Once the cameras have been fully determined the matches can be reconstructed through triangulation. The optimal method for this is given in [11]. This gives us a preliminary reconstruction.

Updating Motion and Structure Recovery To obtain the matrix **P** or the corresponding motion of the camera for all other images in the video sequence a different strategy is used than the one described in the previous section.

First we take an image for which the corresponding matrix **P** has already been computed and retrieve the 2D-3D matches between corners in that image and the reconstructed 3D structure points. Secondly we take another image of which we only have the corners. With our RANSAC algorithm we compute the matrix **F** and corner matches between both images. Using corner matches between corners in image $k-1$ and image k and matches between corners in image $k-1$ and 3D structure points, we obtain matches between corners in image k and 3D structure points. See figure 1.

Knowing these 2D-3D matches we can apply a similar technique as we used to estimate **F**, to determine **P** taking into account equation (4) and a similar RANSAC algorithm. It is important to notice that the matrix **F** serves no longer to extract matrices **P**, but merely to identify corner matches between different images.

Using the previously reconstructed 3D structure points to determine **P** for the next image, we ensure that this matrix **P** is situated in the same projective frame as all previously reconstructed **P**'s. New 3D structure points can be initialized with the newly obtained matrix **P**. In this way the reconstructed 3D environment which one needs to compute **P** of the next image is updated on each step, enabling us to move all around a real object in a 3D environment if necessary.

In this manner motion and structure can be updated iteratively. However the next image to be calibrated cannot be chosen without care. Suppose one chooses two images between which one wants to determine corner matches. If these images are 'too close' to each other, e.g. two consecutive images in a video sequence, the computation of the matrix **F** and therefore the determination of the corner matches between the two images becomes an ill-conditioned problem. Even if the matches could be found exactly the updating of motion and structure is ill-conditioned as the triangulation of newly reconstructed 3D points is very inaccurate as depicted in figure 2.

We resolved this problem by running through the video sequence a first time to build up an accurate but crude 3D reconstruction of the real environment. Accuracy is obtained by using keyframes which are separated sufficiently from each other in the video sequence. See figure 3. Structure and motion are extracted for these keyframes. In the next step each unprocessed image is calibrated using corner matches with the two keyframes between which it is positioned in the video sequence. For these new images no new 3D structure points are reconstructed as they will probably be ill-conditioned due to the closeness of the new image under scrutiny and its neighbouring keyframes. In this way a crude but accurate 3D structure is built up in a first pass along with the calibration of the keyframes. In a second pass, every other image is calibrated using the 2D-3D corner matches it has with its neighbouring keyframes. This leads to both a

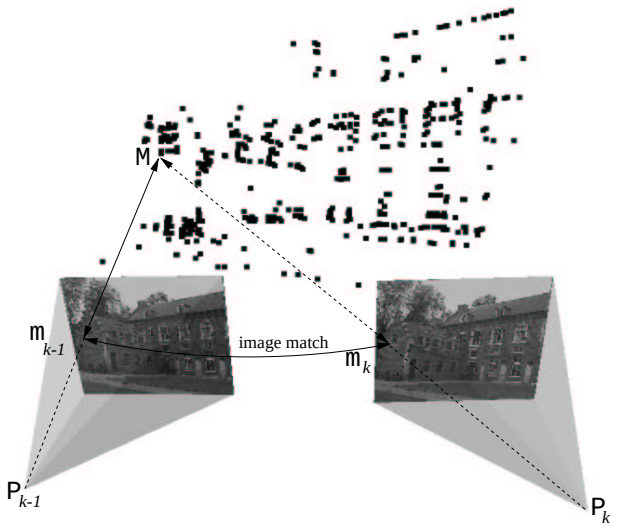

Fig. 1. Knowing the corner matches between image k-1 and image k (m_{k-1}, m_k) and the 2D-3D matches for image k-1 (m_{k-1}, M), the 2D-3D matches for image k can be deduced (m_k, M).

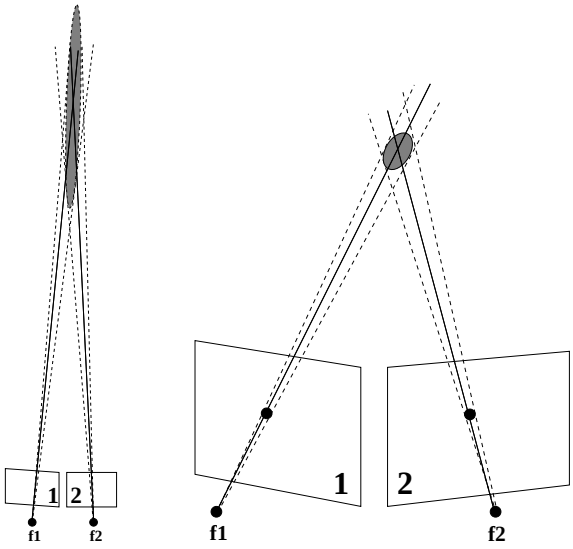

Fig. 2. left: If the images are chosen too close to each other the position and orientation of the camera hasn't changed much. Uncertainties in the image corners lead to a large uncertainty ellipsoid around the reconstructed point. Right: If images are taken further apart the camera position and orientation may differ more from one image to the next, leading to smaller uncertainty on the position of the reconstructed point.

robust determination of the reconstructed 3D environment and the calibration of each image within the video sequence.

Fig. 3. The small dots on the background represent the recovered crude 3D environment. The larger dark spots represent camera positions of keyframes in the video stream. The lighter spots represent the camera positions of the remaining frames.

Metric Structure and Motion Even for an uncalibrated camera some constraints on the intrinsic camera parameters are often available. For example, if the camera settings are not changed during recording, the intrinsic parameters will be constant over the sequence. In general, there is no skew on the image, the principal point is close to the center of the image and the aspect ratio is fixed (and often close to one). For a metric calibration the factorization of the **P**-matrices should yield intrinsic parameters which satisfy these constraints.

Self-calibration therefore consists of finding a transformation which allows the **P**-matrices to satisfy as much as possible these constraints. Most algorithms described in the literature are based on the concept of the absolute conic [8, 24, 18].

The presented approach uses the method described in [18]. The absolute conic ω is an imaginary conic located in the plane at infinity Π_∞. Both entities are the only geometric entities which are invariant under all Euclidean transformations. The plane at infinity and the absolute conic respectively encode the affine and metric properties of space. This means that when the position of Π_∞ is known in a projective framework, affine invariants can be measured. Since the absolute conic is invariant under Euclidean transformations its image only depends on the intrinsic camera parameters (focal length,

...) and not on the extrinsic camera parameters (camera pose). The following equation applies for the dual image of the absolute conic:

$$\omega_k^* \propto \mathbf{K}_k \mathbf{K}_k^\top \tag{6}$$

where $\mathbf{K}_k$ is an upper triangular matrix containing the camera intrinsics for image k. Equation (6) shows that constraints on the intrinsic camera parameters are readily translated to constraints on the dual image of the absolute conic. This image is obtained from the absolute conic through the following projection equation:

$$\omega_k^* \propto \mathbf{P}_k \Omega^* \mathbf{P}_k^\top \tag{7}$$

where Ω^* is the dual absolute quadric which encodes both the absolute conic and its supporting plane, the plane at infinity. The constraints on ω_k^* can therefore be back-projected through this equation. The result is a set of constraints on the position of the absolute conic (and the plane at infinity).

Our systems first uses a linear method to obtain an approximate calibration. This calibration is then refined through a non-linear optimization step in a second phase. More details on this approach can be found in [17].

3 Augmented Video

3.1 Virtual Object Embedding

Results obtained in the previous section can be used to merge virtual objects with the input video sequence. One can import the final calibration of each single image of the video sequence and the reconstructed crude 3D environment into a Computer Graphics System to generate augmented images.

In a Computer Graphics System virtual cameras can be instantiated which correspond to the retrieved calibrations of each image. The image calibrations include translation, rotation, focal length, principal point and skew of the actual real camera that took the image at that time. Typically Computer Graphics Systems do not support skew of the camera. This can easily be adapted in the software of the Computer Graphics System by including a skew transformation after performing the typical perspective transformation as explained in [13]. We use VTK [21] as our Computer Graphics Package. The virtual cameras can now be used to create images of virtual objects.

These virtual objects need to be properly registered with the real 3D environment. This is achieved in the following manner. First virtual objects are placed roughly within the 3D environment using its crude reconstruction. Finetuning of the position is achieved by viewing the result of a rough positioning by several virtual cameras and overlaying the rendering results from these virtual cameras on their corresponding real images in the video sequence. See figure 4. Using specific features in the real video images that were not reconstructed in the crude 3D environment a better and final placement of all virtual objects can be obtained. Note that at this stage of the implementation we don't take into account occlusions when rendering virtual objects.

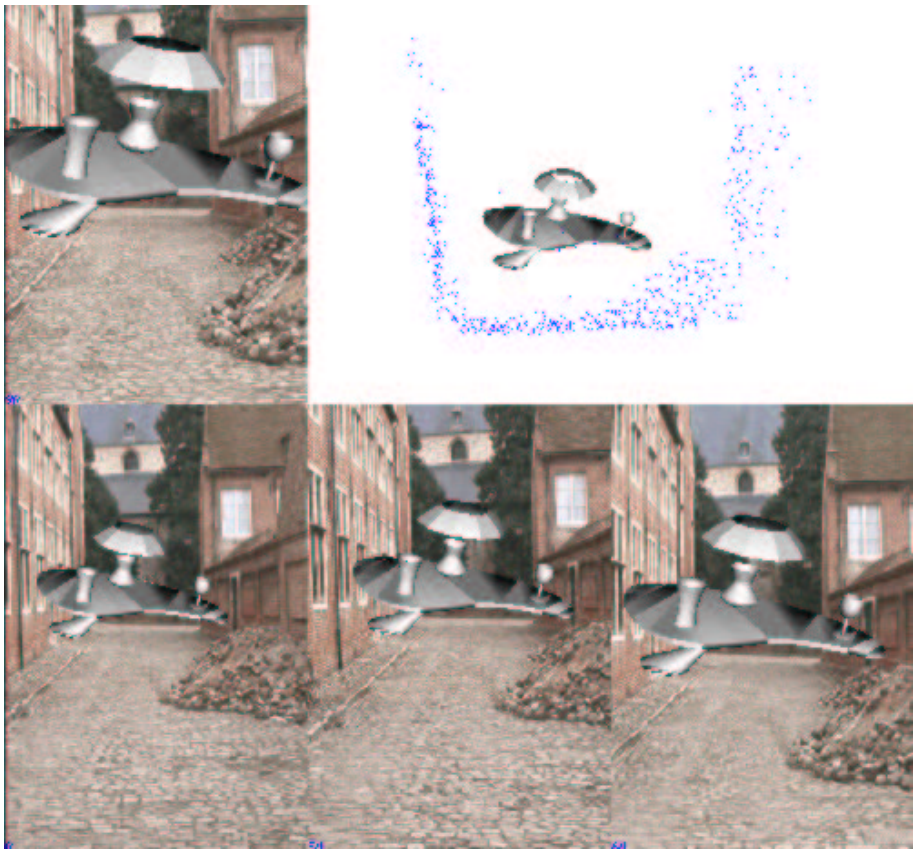

Fig. 4. The AR-interface : In the top right the virtual objects can be roughly placed within the crude reconstructed 3D environment. The result of this placement can be viewed instantaneously on some selected images.

3.2 Virtual Object Merging

After satisfactory placement of each single virtual object the virtual camera corresponding to each image is used to produce a virtual image. The virtual objects are rendered against a background that consists of the original real image. By doing so the virtual objects can be rendered with anti-aliasing techniques using the correct background for mixing.

4 Algorithm Overview

In this section the different steps taken by our AR-System are summarized :

step 1 : The initialization step. Take two images from the video sequence to initialize a projective frame in which both motion and structure will be reconstructed. During this initialization phase both images are registered within this frame and part of the 3D environment is reconstructed. One has to make sure these images are not taken too close or too far apart as this will lead to ill conditions. This is done by imposing a maximum and a minimum separation(counting number of frames) between the two images. The first image pair conforming to these bounds that leads to a good **F**-matrix is selected.
step 2 : Take the last image processed and another image further into the video sequence that still needs registering. Again these images are taken not too close or too far apart with the same heuristic method as applied in step 1.
step 3 : Corner matches between these images and the 2D-3D matches from the already processed image are used to construct 2D-3D matches for the image being registered.
step 4 : Using these new 2D-3D matches the matrix **P** for this image can be determined.
step 5 : Using **P** new 3D structure points can be reconstructed for later use.
step 6 : If the end of the video sequence is not reached, return to step 2.

Now only keyframes that are quite well separated have been processed. The remaining frames are processed in a manner similar to step 3 and 4.

step 7 : For each remaining frame the corner matches of the keyframes between which it lies and their 2D-3D matches are used to obtain 2D-3D matches for this frame.
step 8 : Similar to step 4, the matrix **P** of these frames can be calculated. However no additional 3D structure points are reconstructed.

Now all frames are registered and virtual objects can be placed into the real environment as described in section 3.

step 9 : First the virtual objects are roughly placed within the real environment using its crude 3D reconstruction obtained in previous steps.
step 10 : Finetuning of the positions of the virtual objects is done by seeing the result overlaid on some selected images and adjusting the virtual objects until satisfactory placement is obtained.

5 Examples

We filmed a sequence of a pillar standing in front of our department. Using the AR-System we placed a virtual box on top of this pillar. Note that by doing so we didn't have to solve the occlusion problem for now as the box was never occluded since we were looking down onto the pillar. The AR-System performed quite well. The 'jittering' of the virtual box on top of the pillar is still noticeable but very small. See figure 5.

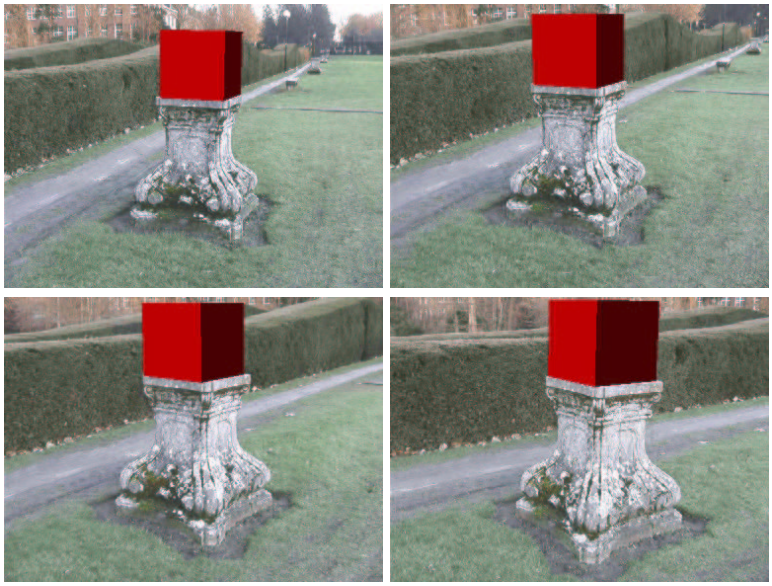

Fig. 5. A virtual box is placed on top of a real pillar. 'Jittering' is still noticeable in the augmented video sequence but is very small.

Another example shows a walk through a street. The camera motion of the person taking the film was far from smooth. However the AR-System managed to register each camera position quite well. See figure 6.

A third example shows another street scene but with a person walking around in it. Despite this moving real object the motion and structure recovery algorithm extracted the correct camera motion. See figure 7.

All video examples can be found at
http://www.esat.kuleuven.ac.be/~kcorneli/smile2.

6 Future Research

It is clear that the proposed AR-System can be further enhanced. One can try to reduce the 'jittering' of virtual objects by considering different techniques. E.g. incorporation

Fig. 6. A street scene: The virtual box seems to stay firmly in place despite the jagged nature of the camera trajectory.

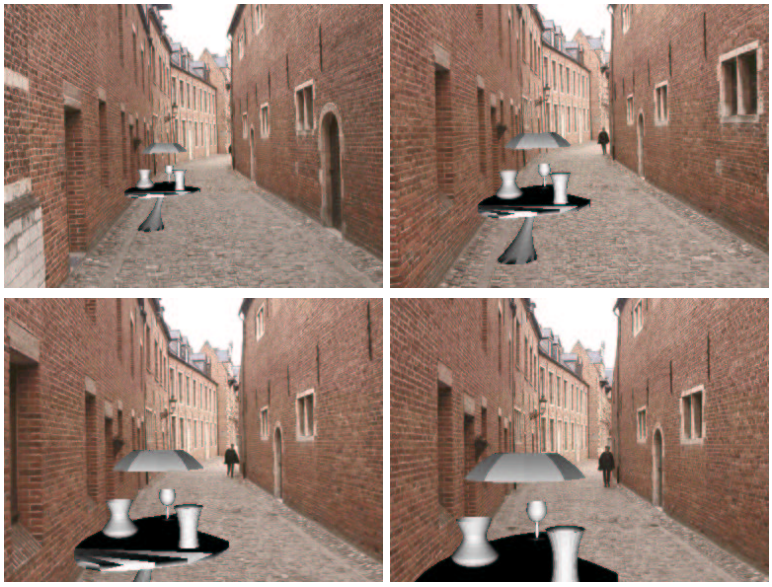

Fig. 7. Another street scene: Despite the moving person the motion of the camera can be extracted and used for augmenting the real environment with virtual objects.

of restrictions on the path followed by the real camera can be used to obtain a smoother path outlined by the virtual cameras. This leads to a smoother motion of the virtual objects in the augmented video and can therefore give more appealing results than the abrupt jumps in motion of noisy virtual camera positions. Another approach to reduce 'jittering' uses real image information in the neighbourhood of the virtual objects to lock it onto a real object. The latter technique is not useful in the case when virtual objects are meant to fly, float or move around in the real environment.

The virtual objects used to augment the real environment can be the result of an earlier 3D reconstruction of real objects. A real vase could be modeled in a first 3D reconstruction step and the result used as virtual object to be placed on top of the real pillar. In this way expensive or fragile objects don't need to be handled physically to obtain the desired video. One can just use its 3D model instead and place it anywhere one wants in a real environment. E.g. relics or statues presently preserved in musea can be placed back in their original surrounding without endangering the precious original. This can be applied in producing documentaries or even a real-time AR-System at the archaeological site itself.

After the registration problem is solved in a satisfactory way we will dive into the occlusion and illumination problems which are still left to be solved and prove to be very challenging.

A topic which seems interesting is to simulate physical interactions between real and virtual objects. A simple form may be to implement a collision detection algorithm which can help us when placing virtual objects onto a surface of the real environment for easy positioning of the virtual objects.

7 Conclusion

In this paper we presented an AR-System which solves the registration problem of virtual objects into a video sequence of a real environment. It consists of two main parts.

The first part tries to recover motion and structure from the images in the video sequence. This motion and structure can be projective but is upgraded to metric by self-calibration. In this way the registration of the virtual objects in the scene is reduced from 15 to 7 degrees of freedom. The second part uses the results of the first part to configure a Computer Graphics System in order to place virtual objects into the input video sequence.

The input to the AR-System is a video sequence which can be totally uncalibrated. No special calibration frames or fiducial markers are used in the retrieval of motion and structure from the video sequence. Also the video sequence does not have to be one of a purely static real environment. As long as the moving parts in the video sequence are small the motion and structure recovery algorithm will treat these parts as outliers(RANSAC) and therefore will discard them correctly in the determination of motion and structure. The Computer Graphics System used for rendering the virtual objects is adapted to use general cameras that include skew of image pixels.

The present AR-System is far from complete. Future research efforts will be made to solve occlusion and illumination problems which are common in Augmented Reality.

Acknowledgements

We wish to gratefully acknowledge the financial support of the ITEA99002 BEYOND project (performed with the support of the flemish district - IWT) and the FWO project G.0223.01.

References

1. R. Azuma: *"A Survey of Augmented Reality."* ACM SIGGRAPH '95 Course Notes No. 9 - Developing Advanced Virtual Reality Applications, (August 1995)
2. P. Beardsley, P. Torr and A. Zisserman: *"3D Model Acquisition from Extended Image Sequences"* Computer Vision - ECCV'96, Lecture Notes in Computer Science, Vol. 1065, Springer-Verlag, pp. 683-695, 1996
3. M.-O. Berger: *"Resolving Occlusion in Augmented Reality: a Contour Based Approach without 3D Reconstruction."*
4. M.-O Berger, G. Simon, S. Petitjean and B. Wrobel-Dautcourt: *"Mixing Synthesis and Video Images of Outdoor Environments: Application to the Bridges of Paris."* ICPR'96, pp. 90-94, 1996
5. D. E. Breen, R. T. Whitaker and E. Rose: *"Interactive Occlusion and Collision of Real and Virtual Objects in Augmented Reality."* Technical Report ECRC-95-02, ECRC, Munich, Germany, 1995
6. P. Debevec: *"Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography."* Proceedings SIGGRAPH 98, pp. 189-198, July 1998
7. P. Debevec, Y. Yu, and G. Borshukov: *"Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping."* Technical Report UCB//CSD-98-1003, University of California at Berkeley, 1998
8. O. Faugeras, Q.-T. Luong and S. Maybank: *"Camera self-calibration: Theory and experiments"* Computer Vision - ECCV'92, Lecture Notes in Computer Science, Vol. 588, Springer-Verlag, pp. 321-334, 1992
9. C. Harris and M. Stephens: *"A combined corner and edge detector"* Fourth Alvey Vision Conference, pp. 147-151, 1988
10. R. Hartley: *"Cheirality invariants"* Proc. D.A.R.P.A. Image Understanding Workshop, pp. 743-753, 1993
11. R. Hartley and P. Sturm: *"Triangulation"* Computer Vision and Image Understanding, 68(2):146-157, 1997
12. M. Fischler and R. Bolles: *"RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography"* Commun. Assoc. Comp. Mach., 24:381-95, 1981
13. J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes: *"Computer Graphics: principles and practice."* Addison-Wesley, Reading, Massachusetts, 1990
14. A. Fournier: *"Illumination Problems in Computer Augmented Reality."* In Journée INRIA, Analyse/Synthèse d'Images (JASI'94), pp. 1-21, January 1994
15. P. Jancène, F. Neyret, X. Provot, J. Tarel, J. Vézien, C. Meilhac and A. Verroust: *"RES: Computing the Interactions between Real and Virtual Objects in Video Sequences."*
16. K. N. Kutulakos and J. Vallino: *"Affine Object Representations for Calibration-Free Augmented Reality."* IEEE Virtual Reality Annual International Symposium (VRAIS), pp. 25-36, 1996

17. M. Pollefeys: *"Self-calibration and metric 3D reconstruction from uncalibrated image sequences"* Ph.D. thesis, ESAT-PSI, K.U.Leuven, 1999
18. M. Pollefeys, R. Koch and L. Van Gool: *"Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters"* International Journal of Computer Vision, Vol.32, Nr.1, pp. 7-25
19. I. Sato, Y. Sato and K. Ikeuchi: *"Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene."* IEEE Transactions on Visualization and Computer Graphics, Vol.5, No. 1, January-March 1999
20. I. Sato, Y. Sato and K. Ikeuchi: *"Illumination Distribution from Brightness in Shadows: Adaptive Estimation of Illumination Distribution with Unknown Reflectance Properties in Shadow Regions."* Proceedings of IEEE International Conference on Computer Vision (ICCV'99), pp. 875-882, September 1999
21. W. Schroeder, K. Martin and B. Lorensen: *"The Visualization Toolkit 2nd edition."* Prentice Hall, New Jersey, 1998
22. C. Schütz and H. Hügli: *Augmented Reality using Range Images."* SPIE Photonics West, The Engineering Reality of Virtual Reality 1997, San Jose, 1997
23. Y. Seo, M. H. Ahn and K. S. Hong: *"Video Augmentation by Image-based Rendering under the Perspective Camera Model."*
24. B. Triggs: *"The Absolute Quadric"* Proc. 1997 Conference on Computer Vision and Pattern Recognition, IEEE Computer Soc. Press, pp. 609-614, 1997
25. M. Tuceryan, D. S. Greer, R. T. Whitaker, D. Breen, C. Crampton, E. Rose and K. H. Ahlers: *"Calibration Requirements and Procedures for Augmented Reality."* IEEE Transactions on Visualization and Computer Graphics, pp. 255-273, Sept 1995
26. T. Tuytelaars and L. Van Gool: *"Content-based Image Retrieval based on Local Affinely Invariant Regions."* Third International Conference on Visual Information Systems, Visual99, pp. 493-500, june 1999
27. M. Uenohara and T. Kanade: *Vision-Based Object Registration for Real-Time Image Overlay."* Proceedings CVRMED'95, pp. 14-22, 1995
28. M. Wloka and B. Anderson: *"Resolving Occlusion in Augmented Reality."* ACM Symposium on Interactive 3D Graphics Proceedings, pp. 5-12, April 1995

Discussion

1. **Kostas Daniilidis, University of Pennsylvania**: When you have a simple task, like in your case inserting a cube, it is not necessary to compute a Euclidean reconstruction. There is other work, see Kutulakos and Vallino [1], which describes systems that just assume scaled orthographic projections.
   Kurt Cornelis: I think that's true, but we are also aiming at building a Euclidean reconstruction with which we can finally interact in a way that we are used to in the real world. We might want to compute a trajectory of an object in a real-life manner. I don't see how you can easily calculate the equivalent trajectory in a projective reconstruction. We are thinking now of future applications, so we want to obtain a Euclidean reconstruction in advance.
   Marc Pollefeys: It is simpler to put a virtual object in the scene when the metric structure is available. In this case only 7 parameters (corresponding to a similarity transformation) have to be adjusted, while for insertion in a projective reconstruction 15 parameters need to be adjusted. Some of these parameters are not as intuitive to adjust as rotations, translation and scale. So if the information for a metric upgrade is in the images, it is better to take advantage of it.
2. **Kyros Kutulakos, University of Rochester**: I definitely agree with you that Euclidean reconstruction is very important. But I think you should distinguish between augmented reality systems where the input is live video, real-time, and systems where you are working on recorded video. I'm wondering if you could comment on how easy it would be to do this in real-time?
   Kurt Cornelis: The system has been designed for off-line processing of recorded video. The computational requirements to deal with hand-held markerless video data exceed the capabilities of real-time systems. Furthermore the current implementation is working with keyframes and relies on their availability from the start. The proposed approach would thus not be simple to adapt to work with real-time video streams.
3. **Andrew Fitzgibbon, University of Oxford**: You note that jitter is low, but in a system such as this, you wouldn't expect to get jitter because you are fitting into the image. However, you would expect to get drift because errors are being accumulated over time. To what extent is drift an issue?
   Kurt Cornelis: I haven't really considered drift. The video sequences you saw were actually quite short. So I think there was not enough time to experience drift. I think it is good to investigate this for longer sequences and see what it gives. Thank you for the comment.
4. **Richard Szeliski, Microsoft**: It was interesting to hear that you thought you had to model skew. You couldn't live with a computer graphics package that didn't allow that. I thought I heard the other speakers say that we agree the skew is zero for all practical purposes. That's why I wanted to hear your comment.
   Kurt Cornelis: As I said, the metric update is not going to be perfect. The cameras obtained after this update are still going to have some small skew and we want to be able to model this.
   Marc Pollefeys: We plan to put a bundle adjustment in the system and enforce the skew to be zero. This was just a first implementation and it was easier to twist VTK

to handle skew than to implement bundle adjustment just to see if the system is working well. If zero-skew is enforced without bundle adjustment, it will introduce jitter in the augmented video, because the projection matrices are modified without taking the effect on the reprojection error into account. The metric reconstruction can be off by a few degrees compared to the true scene but this is in general not visible. Note that bundle adjustment will probably reduce this error, but there will always be some error.

5. **Jean Ponce, University of Illiniois at Urbana-Champaign**: Concerning projective versus metric reconstruction, I think it depends on the application. For example, with your cube that you are placing against the wall, you can just put some markers on the wall and track them. They form a natural way to do the interface. But maybe for medical application like surgery, a metric reconstruction is more needed.
   Kurt Cornelis: I totally agree, it depends on the application at hand.
6. **Kyros Kutulakos, University of Rochester**: I don't think I agree with Jean or Kostas, you can certainly put objects in the scene projectively but you cannot do shading projectively. So unless you want to render images where you have surfaces that have flat texture, which was what I did, rendering a mirroring sphere would be very hard to do projectively.
   Andrew Zisserman, University of Oxford (comment): After auto-calibration there may be some slight residual projective skew in 3D (between the reconstruction and ground-truth). The effect of this is that objects inserted into the images will have a slight skew, but this might not be very noticeable. The same with lighting errors, a small error in the normals because of projective skew may not be very noticeable.
   Marc Pollefeys: In this case I do not fully agree with Kyros. We certainly need metric structure to get the lighting and other things correct, but by correctly inserting a virtual object (which is a metric object) in a projective reconstruction we do in fact carry out a calibration.
   Kostas Daniilidis, University of Pennsylvania: I talked about affine and not projective reconstruction which comes to what Rick Szeliski indicated earlier—that we should establish some metrics for the people for whom we are going to solve these things, whether affine reconstruction is important, whether the drift is important, whether the jittering is the most important aspect? This would be nice to quantify somehow.

References

1. K. Kutulakos and J. Vallino. Calibration-free augmented reality. *IEEE Transactions on Computer Graphics*, 4(1):1–20, 1998.