A Survey of Online Monocular Markerless Augmented Reality

Veronica Teichrieb[1], João Paulo Silva do Monte Lima[1], Eduardo Lourenço Apolinário[1], Thiago Souto Maior Cordeiro de Farias[1] Márcio Augusto Silva Bueno[1], Judith Kelner[1], and Ismael H. F. Santos[2]

[1]Computer Science Center
Federal University of Pernambuco - Recife-PE

[2]CENPES Petrobras, Rio de Janeiro, RJ

{vt, jpsml, ela, mouse, masb, jk}@cin.ufpe.br, ismaelh@petrobras.com.br

***Abstract*—This paper surveys the field of Markerless Augmented Reality, specifically online and monocular. This research field is applied by the TechPetro project that aims to define and developed a Markerless Augmented Reality framework for the implementation of Augmented Reality based engineering solutions. In Markerless Augmented Reality, 3D virtual objects are integrated into a 3D real environment in real-time. This is achieved using the world as marker instead of fiducial markers applied in traditional Augmented Reality systems. It discusses major issues related to the field, such as tracking and registration, which become much more complex. This paper also describes the characteristics and experimental results of online monocular Markerless Augmented Reality techniques. Future directions and areas requiring further research are briefly discussed. This survey provides a starting point for anyone interested in researching or using Markerless Augmented Reality.**

***Index Terms*—Markerless Augmented Reality, Online and monocular techniques, Survey.**

I. INTRODUCTION

This paper surveys the current state-of-the-art in Markerless Augmented Reality, from now on named MAR throughout this paper. It describes work performed at many different sites and explains the issues and problems approached related to online and monocular MAR systems. It summarizes the tradeoffs and approaches taken so far to overcome problems and speculates on some future directions that deserve exploration.

A survey paper does not present new research results. The contribution comes from consolidating existing information from many sources and publishing an extensive bibliography of papers in this field. This paper provides a good starting point for anyone interested in beginning research in this area.

Section 1 describes what MAR is, and contextualizes the authors' studies in the TechPetro project. Section 2 explains the main techniques developed for building online monocular MAR systems. Finally, Section 3 draws some conclusions, highlighting thoughts regarding MAR tendencies, and describing some areas that require further work and research.

See http://www.cin.ufpe.br/˜grvm
Manuscript received June 30, 2007; revised July 20, 2007.

A. Definition

MAR systems integrate 3D virtual objects into a 3D real environment in real-time, enhancing user's perception of and interaction with the world. Its basic difference from marker based AR systems is the method used to place virtual objects in the real world. This approach is not based on the use of traditional artificial markers that need to be placed in the world to be tracked by the system in order to calculate their position and orientation.

In MAR any part of the real environment may be used as a marker that can be tracked in order to position virtual objects. Therefore, there are no ambient intrusive markers that are not really part of the environment. Furthermore, MAR counts on specialized and robust trackers. Another advantage is the possibility of extracting from the environment characteristics information that may later be used by the MAR system.

Nonetheless, tracking and registration techniques become more complex in MAR systems. Another disadvantage emerges in online MAR since it presents more restrictions.

B. The TechPetro Project

MAR technology has been studied in the context of the TechPetro project. TechPetro is a two years project, developed by the Virtual Reality and Multimedia Research Group (GRVM) in association with CENPES Petrobras and FINEP. In this project, engineering solutions will be developed based on two technologies: MAR and 3D reconstruction from 2D images. These technologies allow the automatic 3D reconstruction of complex scenes captured from the real world, as well as the augmentation of user's perception through the use of an interface that integrates in real-time 3D virtual information into the real world scene visualized by the user.

This paper is related to TechPetro's MAR studies. In sequence, MAR techniques are classified, described and compared.

II. MAR TECHNIQUES

Techniques developed for MAR can be classified in two major types: model based and Structure from Motion (SfM) based (Figure 1). In model based techniques, knowledge about

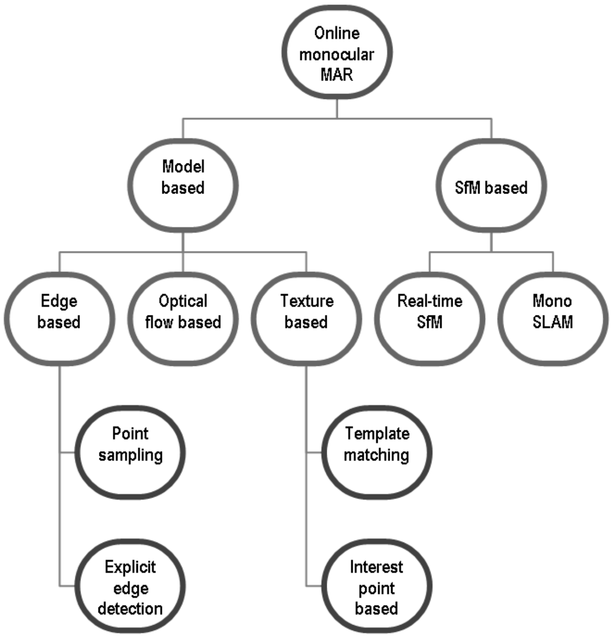

Fig. 1. Online monocular MAR taxonomy.

the real world is obtained before tracking occurs and is stored in a 3D model that is used for estimating camera pose. In SfM based approaches, camera movement throughout the frames is estimated without any previous knowledge about the scene, which is acquired during tracking. Model based methods are often simpler than SfM based ones, but tracking depends on the visibility of the previously modeled objects in the real world image. SfM based systems do not have this constraint, since they are capable of continuously tracking the camera egomotion in unknown scenes. Both MAR techniques types are detailed in the following subsections.

A. Model Based

One of the approaches used to calibrate the camera in MAR applications relies on the use of 3D Computer Aided Design (CAD) models relative to the objects present in the real scene [1]. 3D models are matched with the 2D image of the real world and the output of this process is the objects that are present in the image, their position and orientation. If there is not a 3D model of the object to be tracked, then it can be obtained using any 3D digitizing method, e.g. optical 3D reconstruction. This process is almost always offline and consists in a preparation stage to the online phase of the tracker, where the model is correlated with the image.

The advantage of using a model based approach is the possibility of interaction between real and virtual worlds, like occlusion and collision [2], as can be seen in the examples illustrated in Figure 2. In order to accomplish such types of interaction, the application exploits the fact that the real object pose is known and its structure is described by the 3D model. The 3D model is utilized in the physics simulation and the visibility algorithm, but it is not overlaid onto the image (only the remaining virtual elements are).

Fig. 2. A virtual car colliding with a real castle ruin (left) and being occluded by a real castle (right) [2].

Model based approaches require an a priori knowledge about the real scene, since 3D models of the objects are needed. Due to the offline model acquiring process, these techniques are, in general, not totally online and cannot be used in unprepared environments. Furthermore, tracker initialization is usually done manually or requires a prior training in order to be automatic. This aspect is critical for AR applications, since any tracking failure will force the user to reinitialize the system by hand. Another issue is related to the fact that the tracked objects need to be present all the time in the image, otherwise it will not be possible to retrieve the camera pose and augment the scene.

Model based techniques can be classified in three categories. The first category consists in methods that take only the objects' edges into consideration while doing tracking [2] [3]. The second one relies on the optical flow of the image sequence [4], while the third one comprises the use of objects' texture information to perform tracking [5] [6]. Aspects related to each category are described next.

1) Edge Based: In this category, camera pose is estimated by matching a wireframe 3D model of an object with the real world image edge information. This matching is achieved by projecting the model onto the image and minimizing the displacement between the projected model and the imaged object. In order to accomplish this task, a good initial hint about the object pose is needed. In edge based methods, the initialization is done manually. Once the first pose is estimated, it is used to project the model onto the next frame. Assuming that camera displacement between consecutive frames is relatively small, using the previous pose estimation to predict the next pose will not harm the matching process.

Edge based techniques were the first approaches to real-time 3D object tracking [1]. Due to their low complexity, they are easy to implement and have a good performance. Because they only use edge information, edge based approaches are able to track specular objects affected by environment lighting. However, edge based methods usually do not support fast camera movements, since the projected model will be too far from its correct location. Another problem is related to matching errors, which may be caused by elements such as a cluttered background or shadows in the image.

Edge based methods can be divided in two subcategories. The first subcategory comprises methods that sample some control points along the edges of the wireframe 3D model and compare their projections with strong gradients present in

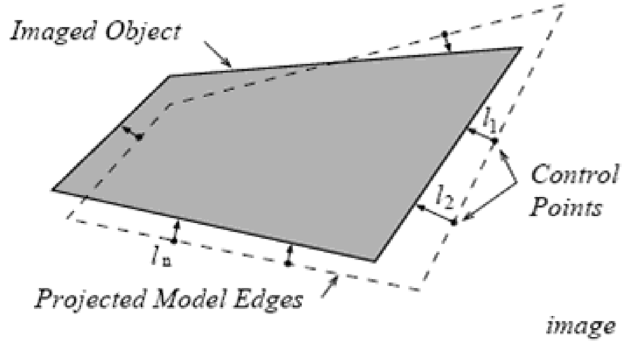

Fig. 3. Matching between the projected control points of the model and the imaged object [7].

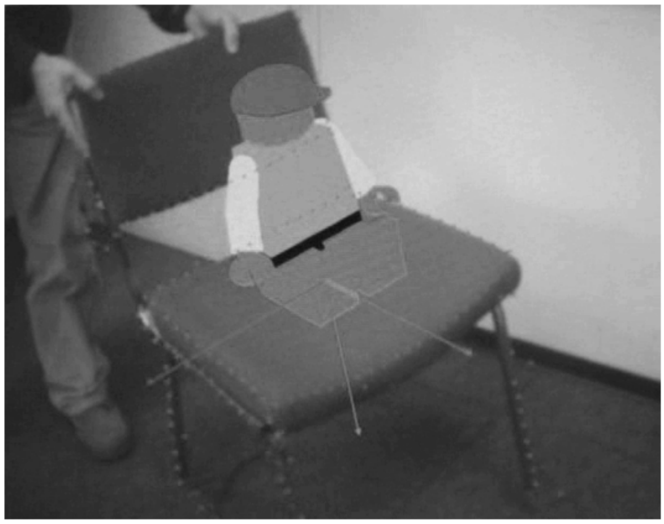

Fig. 4. Point sampling MAR application [2].

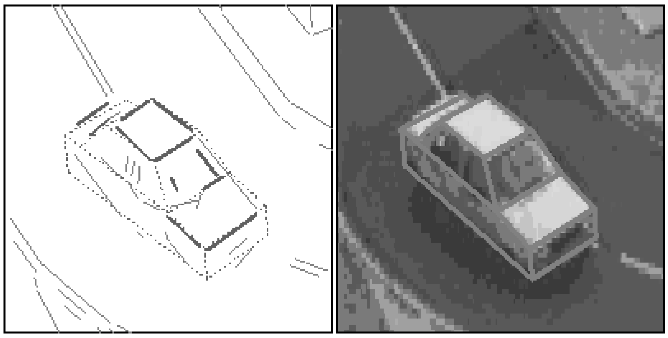

Fig. 5. Detected lines on the image (left) and pose estimated by matching with the projected model (right) [3].

the image [2]. The second subcategory encloses methods that detect explicit edges on the image and match them with the model projection [3].

The first task performed by point sampling methods is choosing which control points are going to be considered in the matching phase. Based on the previous estimated pose, the visible edges of the model are determined. Only the control points that belong to a visible edge are going to be used. After such selection, the points are projected onto the image plane using previous camera pose estimation. Image edges are detected by calculating the gradients in both x and y directions. Then, there is a search around the neighborhood of the projected points in order to find the corresponding points in the image edges [7] (see Figure 3). Finally, the camera relative motion is estimated based on the variation of the points' positions.

Point sampling methods are very efficient, since the processing involved is rather simple. They are also very general, as they can cover curved edge objects. The disadvantage of point sampling approaches is their lack of robustness in the matching phase, which can lead to incorrect pose estimation and jittering. This problem may be addressed using robust estimators, insensitive to noise sources, like occlusions and background cluttering. An example of a MAR application where a lego toy is placed on a chair is shown in Figure 4 and described in [2]. The control points and the chair coordinate system can also be distinguished.

In the explicit edge detection technique, edges are extracted from the image using a line detection operator such as the Hough transform. The wireframe model is projected onto the image using the previously estimated camera pose. The projection and the image edges are compared in order to calculate the current camera pose [3]. Figure 5 illustrates the process.

Explicit edge detection methods are more robust than point sampling ones, but lack in generality, since the usage of lines restricts its use to polygonal objects. In addition, point sampling methods are more efficient than explicit edge detection methods.

2) Optical Flow Based: Differently from edge based methods, which rely on spatial information obtained by image-model matching, optical flow based tracking exploits temporal information. This is extracted from the relative movement of the object projection onto the image. After initialization, which is often manual, the optical flow between the frames captured at time t and t+1 is calculated. Then, the algorithm determines which points from the model projected onto the image at time t are still present in the image at time t+1. The displacement of these points over time is calculated using an algorithm such as the Kanade-Lucas (KL), described in [8]. This is used to estimate camera movement.

Due to its integration over time, 3D tracking based on optical flow presents smoother changes between consecutive poses. Another advantage is the moderate processing load needed. However, optical flow techniques tend to accumulate errors produced by sequential pose estimations, leading to a deviation from the correct camera calibration. Optical flow algorithms are also not robust against lighting changes and large camera displacements, originating errors in object tracking and requiring re-initialization. Figure 6 shows an optical flow based application that performs 3D face tracking [4].

3) Texture Based: In this section, texture based techniques will be presented, together with their general principles. As the name says, this category of techniques takes into account

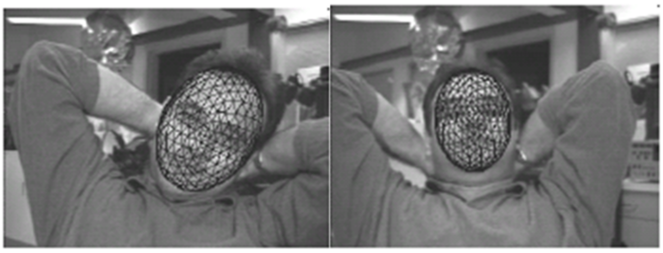

Fig. 6. Optical flow based MAR application [4].

Fig. 7. 3D tracking with template matching - template image in the center of the marker (left) and augmented scene (right) [11].

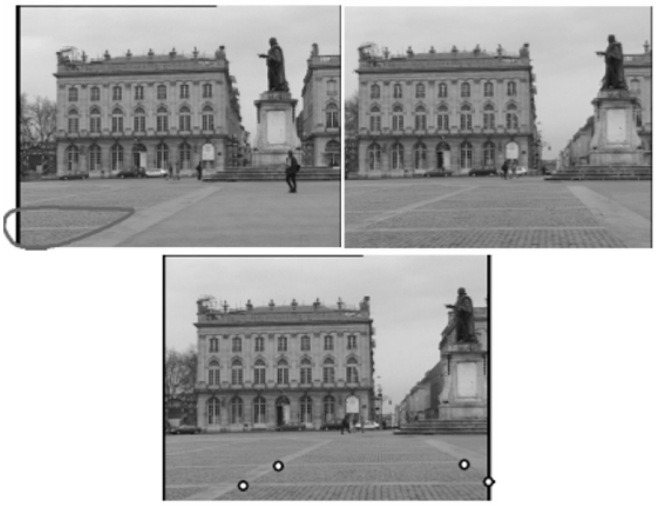

Fig. 8. Algorithm steps - patch manually selected (top left), automatic feature detection (top right) and tracking plane (bottom) [14].

texture information presented in images. Firstly, template matching techniques will be described [5]. This subcategory has been proposed in order to apply a distortion model to a reference image to recover rigid object movement. Another subcategory is interest point based [6], reasonably similar to the optical flow technique, that is, they also use just local features but taking into consideration texture information to help the search and tracking.

The template matching approach is based on global information, unlike feature based techniques. The strength of this subcategory lies in its ability to treat complex patterns that would be difficult to model by local features. These techniques are also called sum-of-square-difference or SSD, as they consist in minimizing the difference between a region of the image and a reference template.

Basically, such techniques search for the parameters of a function that warps a template into the target image, so that tracking can be done. According to [8], this is the general goal of the KL algorithm. In [9], the author shows an approach based on the Jacobian of the warping function used in the KL algorithm to do 2D tracking. However, there are some problems with variations in illumination and partial occlusions.

When it comes to 3D tracking, the Jacobian approach shows some difficulties and does not achieve good results. Therefore, in [5], instead of using the inverse of the Jacobian image, one could approximate the variations in pixels intensities using hyperplanes, leading to better results without any further computation. Using the hyperplane approximation, [10] showed how to treat illumination changes, partial occlusion and fast motion, using normalization of lighting changes and an extended motion model. Figure 7 shows a simple augmented scene where an image was used as a template.

The subcategory of interest point based techniques takes into account localized features, instead of a global search used by template matching techniques. As a result, this subcategory is less computer-intensive than former ones. Another advantage is the fact that illumination changes are easily achievable. In [12], the author mentions that as no inter-frame assumption is made, it allows a wider baseline than optical flow.

As previously said, this technique is based on local features selection, and according to [6] these features can be patches manually selected in a preliminary stage. As there is human intervention during this step of the process, the selection of patches demands specific expertise, and it would be very interesting that these patches could be automatically selected; there comes the main idea of interest point techniques.

It is possible to match only a subset of the image, using an interest operator. Basically, the interest operators must select all points with a certain set of characteristics. In [1], there is a good survey about these operators.

Tracking can be done using KL [8]. An alternative to matching is achieved through the use of the Kanade-Lucas-Tomasi (KLT) tracker [13]. Initially, there is an extraction phase, where features with higher eigenvalues are selected. After that, the tracking phase relies mostly on KL. One advantage of this approach is that continuity is easier to achieve, but KLT offers some performance issues.

There is also an interest point technique based on tracking planes [14], instead of full 3D models, and it can be seen in Figure 8. The main idea here is to explore the homography formed by the plane in two consecutive views. This computation is performed using the RANdom SAmple Consensus (RANSAC) algorithm, and recursively determines which homography is correct.

As someone could expect, this method accumulates errors during its execution, once there is no a priori knowledge about any points in the scene, occasioning some drift and a minimal jittering. In [15], the concept of keyframes is introduced to resolve the drift issue. Keyframes are images of the scene in which interest points are precalculated. Then, every incoming frame is matched against the closest keyframe. It is easy to see why this approach is not scalable, once huge differences can happen between keyframes and incoming frames, occasioning drift; hence, this approach is hardly applicable to an AR application.

In [15], the author also shows how to use a mix of the keyframes approach and information provided by preceding frames to enforce temporal coherence. This way, the problem of tracking a 3D model is reformulated as a bundle adjustment problem. In [16], there is a simple way of using edge information to make the tracker more robust.

B. SfM Based

Instead of relying on previously obtained information about the scene to be tracked, some MAR techniques estimate the camera displacement without any a priori knowledge about the environment. These methods are also able to retrieve the structure of the scene in real-time, with different levels of detail, depending on the approach used.

SfM based techniques are mainly online, since they do not require any previous offline learning phase. Due to this, it is possible to reconstruct a totally unknown environment on the fly. As a drawback, SfM approaches are often very complex. They also have some constraints related to their real-time nature.

1) Real-Time SfM: A classic technique used in computer vision to make 3D reconstruction is SfM [17]. Its traditional implementation follows a suggested pipeline, which is not concerned with real-time constraints.

SfM produces great results relative to the final mesh generated by the entire process, but some algorithms present in its pipeline require a lot of processing time to finish their work. Usually, the SfM pipeline is composed of the following phases: feature tracking (normally using some optical flow based tracker, like KL [8] or KLT [13]), fundamental matrix extraction and refinement, camera pose estimation and self-calibration.

Basically, in order for SfM to support real-time constraints, some of these phases have to be simplified or replaced by other algorithms that still maintain the robustness of this technique. In Nistr's implementation of real-time SfM, he introduced some modifications to the pipeline relative to the refinement of the points given by the feature tracker, creating a brand new solution based on the classic RANSAC refinement algorithm [18]. He used this new algorithm to classify the points and eliminate outliers, producing a mesh without noise. This algorithm works similar to RANSAC, but in a preemptive way, stopping points classification when a good result is reached.

In addition, camera pose estimation and self-calibration phases were replaced by the five-point method in Nistr's SfM implementation [19]. This method consists in solving a linear equation, considering the number of degrees of freedom given by the metric reconstruction. Therefore, to compute the camera translation and orientation, only five points are used. Another remark is that to achieve good results, the intrinsic camera parameters have to be fixed.

These modifications to the original SfM pipeline removed some bottlenecks and speeded up the entire process, allowing a minimum delay to reconstruct a rigid scene and hence getting closer to real-time 3D reconstruction. Figure 9 illustrates a mesh computed using Nistr's SfM technique.

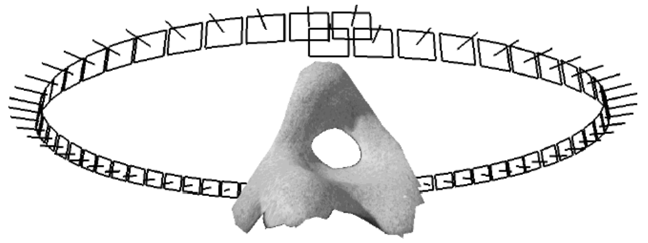

Fig. 9. Example of a mesh computed by the real-time SfM technique, assuming a circular camera trajectory [20].

Fig. 10. Real-time SfM MAR application [20].

Since the real-time constraint is supported by SfM, it has been used in MAR systems. Real-time SfM can offer more information about the entire scene, and may provide data to improve the MAR system with features like occlusion of virtual objects by real ones and physical based interaction between them. Although 3D information about the scene can supply all these results in MAR, it has not been strongly explored yet. Many efforts have been applied to attach SfM to MAR, but until now only some results were achieved [20] (see illustration in Figure 10).

Because SfM is a traditional technique, a lot of internal algorithms applied by it are already implemented in most computer vision libraries [21]. Trackers, linear equation solvers, camera pose estimators and more pieces of code are available to developers, simplifying the creation of a 3D reconstruction system based on SfM. However, a good knowledge about how the technique works internally is mandatory to achieve real-time execution. This is obtained by hacking mathematical equations, optimizing them and removing some variables that are not used after a shortcut in the math flow.

2) MonoSLAM: SLAM (Simultaneous Localization and Mapping) is a well defined and used approach in the robotic community for constructing a representation of the environment on the fly and estimating robot motion. SLAM is

Fig. 11. Active search ellipses (left) and scene with four virtual objects (right) [22].

mainly accomplished by using modern methods of sequential Bayesian inference and normally uses sensors such as laser range-finders and sonar. MonoSLAM was created based on the probabilistic SLAM methodology using a single freely-moving wide-angle camera as the only sensor and with a real-time constraint [22].

The MonoSLAM algorithm runs at 30 frames per second (fps), estimates camera pose and creates a sparse map of the environment natural landmarks. It is a very efficient algorithm with a low level of jitter (1-2 cm) and drift-free, while being robust to handle extreme rotation, occlusion and closed loop. However, it is restricted to indoor environments, smooth camera movement and monochrome camera image.

To initialize the system, a known picture is necessary to be present in the initial frame at an approximated certain distance. The algorithm begins searching for features in the image utilizing the image interest operator of Shi and Tomasi [13] to locate the best candidate within a limited window of 80x60 pixels. This window is randomly positioned in any area that does not contain other features nor will be out of the camera view, based on the current camera and angular velocities.

When good features are selected, the algorithm estimates its depth and associates it with a level of uncertainty. As the feature continues to be tracked in the next frames, the depth estimation is enhanced and the feature is fully initialized and stored as an oriented planar texture of 11x11 pixels. It utilizes the Davison and Murray's approach, which relies on visual landmarks, as they have more unique signatures than standard corner features [23]. The tracking of each feature occurs within an ellipse region, and shape and position in the image are defined based on its level of uncertainty and on the camera estimated movement, respectively (Figure 11, left).

The features are inserted in a probabilistic feature based map that is maintained during all the lifetime of the operation and is updated by the Extended Kalman Filter (EKF). The map grows as new features are added, or shrinks when a feature that fails to be detected many times is removed.

The MonoSLAM technique was used in an AR scenario where virtual furniture is added to an image stream captured by a handheld camera (Figure 11, right). The virtual objects showed to be stable in this example.

III. CONCLUSION

This paper has surveyed MAR techniques that use only one camera and work in real-time. Nevertheless, there are other MAR approaches, such as that in [24], which works offline and is mainly applied to video post production.

Among the previously presented tracking methods for MAR applications, SfM based techniques should be highlighted, due to their ability to augment completely unknown scenes. Research regarding SfM applied to MAR is still in its infancy and therefore there are several open problems that need careful attention. For example, real-virtual interaction, which is exploited by model based MAR applications [2], has not yet been approached by SfM based ones. Indeed, SfM based techniques retrieve information about the environment that could be used to build complex applications taking advantage of such kind of interaction.

However, it is important to say that, according to the problem tackled, purely model based or hybrid approaches should be considered. Even though there are markerless techniques for augmented reality, the use of markers is suitable for systems that do not mind the presence of artificial elements in the scene. In other words, the solution is defined by the kind of problem being faced.

ACKNOWLEDGMENT

The authors would like to thank FINEP and CENPES Petrobras for financially supporting this research (process 3662/2006).

REFERENCES

[1] LEPETIT, V.; FUA, P. Monocular model-based 3d tracking of rigid objects. *Found. Trends. Comput. Graph. Vis.*, Now Publishers Inc., Hanover, MA, USA, v. 1, n. 1, p. 1–89, 2006. ISSN 1572-2740.

[2] COMPORT, A. I. et al. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Computer Society, Los Alamitos, CA, USA, v. 12, n. 4, p. 615–628, 2006. ISSN 1077-2626.

[3] KOLLER, D.; DANIILIDIS, K.; NAGEL, H. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, v. 10, n. 3, p. 257–281, June 1993. Disponível em: <citeseer.ist.psu.edu/koller93modelbased.html>.

[4] BASU, S.; ESSA, I.; PENTLAND, A. *Motion regularization for model-based head tracking*. 1996. Disponível em: <citeseer.ist.psu.edu/article/basu96motion.html>.

[5] JURIE, F.; DHOME, M. A simple and efficient template matching algorithm. In: . [s.n.]. p. 544–549. Disponível em: <citeseer.ist.psu.edu/jurie01simple.html>.

[6] UENOHARA, M.; KANADE, T. Vision-based object registration for real-time image overlay. *Journal of Cognitive Neuroscience*, v. 3, n. 1, p. 71–86, 1991.

[7] ARMSTRONG, M.; ZISSERMAN, A. *Robust object tracking*. 1995. Disponível em: <citeseer.comp.nus.edu.sg/armstrong95robust.html>.

[8] LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision (darpa). In: *Proceedings of the 1981 DARPA Image Understanding Workshop*. [S.l.: s.n.], 1981. p. 121–130.

[9] HAGER, G. D.; BELHUMEUR, P. N. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 10, p. 1025–1039, 1998. Disponível em: <citeseer.ist.psu.edu/hager98efficient.html>.

[10] DELLAERT, F.; COLLINS, R. *Fast image-based tracking by selective pixel integration*. 1999. Disponível em: <citeseer.ist.psu.edu/dellaert99fast.html>.

[11] LEPETIT, V.; LAGGER, P.; FUA, P. Randomized trees for real-time keypoint recognition. In: *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*. Washington, DC, USA: IEEE Computer Society, 2005. p. 775–781. ISBN 0-7695-2372-2.

[12] RAVELA, S. et al. Adaptive tracking and model registration across distinct aspects. In: *IROS95*. [s.n.], 1995. p. 174–180. Disponível em: <citeseer.ist.psu.edu/article/ravela95adaptive.html>.

[13] SHI, J.; TOMASI, C. Good features to track. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*. Seattle: [s.n.], 1994. Disponível em: <citeseer.ist.psu.edu/shi94good.html>.

[14] SIMON, G.; BERGER, M.-O. Pose estimation for planar structures. *IEEE Computer Graphics and Applications*, v. 22, n. 6, p. 46–53, 2002.

[15] VACCHETTI, L.; LEPETIT, V. Stable real-time 3d tracking using online and offline information. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 26, n. 10, p. 1385–1391, 2004. ISSN 0162-8828. Member-Pascal Fua.

[16] VACCHETTI, L.; LEPETIT, V.; FUA, P. Combining edge and texture information for real-time accurate 3d camera tracking. In: *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*. Washington, DC, USA: IEEE Computer Society, 2004. p. 48–57. ISBN 0-7695-2191-6.

[17] POLLEFEYS, M. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. [S.l.], maio 1999.

[18] NISTéR, D. Preemptive ransac for live structure and motion estimation. *Mach. Vision Appl.*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 16, n. 5, p. 321–329, 2005. ISSN 0932-8092.

[19] NISTéR, D. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 26, n. 6, p. 756–770, 2004. ISSN 0162-8828.

[20] LOURAKIS, M.; ARGYROS, A. Efficient, causal camera tracking in unprepared environments. *Computer Vision and Image Understanding*, v. 99, p. 259–290, 2005. Disponível em: <citeseer.ist.psu.edu/lourakis05efficient.html>.

[21] BRADSKI, G. R.; PISAREVSKY, V. Intel's computer vision library: Applications in calibration, stereo, segmentation, tracking, gesture, face and object recognition. *cvpr*, IEEE Computer Society, Los Alamitos, CA, USA, v. 02, p. 796–797, 2000. ISSN 1063-6919.

[22] DAVISON, A. J.; MOLTON, N. D. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 29, n. 6, p. 1052–1067, 2007. ISSN 0162-8828. Member-Ian D. Reid and Member-Olivier Stasse.

[23] DAVISON, A. J.; MURRAY, D. W. Simultaneous localization and map-building using active vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 24, n. 7, p. 865–880, 2002. ISSN 0162-8828.

[24] CORNELIS, K. *From uncalibrated video to augmented reality*. [S.l.], nov. 2004.