MAPPING AND RE-LOCALIZATION FOR MOBILE AUGMENTED REALITY

Pierre Martin[1,2]*, Eric Marchand*[1]

[1]IRISA
Université de Rennes 1
Rennes, France

Pascal Houlier[2]*, Isabelle Marchal*[2]

[2]Orange Labs
Cesson Sevigné, France

ABSTRACT

Using Simultaneous Localization And Mapping (SLAM) methods become more and more common in Augmented Reality (AR). To achieve real-time requirement and to cope with scale factor and the lack of absolute positioning issue, we propose to decouple the localization and the mapping step. We explain the benefits of this approach and how a SLAM strategy can still be used in a way that is meaningful for the end user. The method we proposed has been fully implemented on various smartphone in order to show its efficiency.

Index Terms— Augmented reality, simultaneous location and mapping, mobile phone

1. INTRODUCTION

The goal of augmented reality is to insert virtual information in the real world providing the end-user with additional knowledge about the scene. The added information, usually virtual objects, must be precisely aligned with the real world. It is then necessary to accurately align real and virtual world and then to compute the full position of the device for each image of the sequence. To achieve this goal, camera localization is a key feature of all augmented reality systems.

With the development of powerful smartphone, it becomes possible to foresee AR applications on such devices. Nevertheless most of AR Apps only consider sensors such as IMU, compass and GPS and barely consider image-based localization approaches. Regardless the computational cost of such methods, one of the main reason is that the usually rely on the use of 3D model [1, 2] or assumption either the structure of the scene (supposed to be planar) or the camera motion (supposed to be rotation) [3, 4]. Nevertheless, since the introduction of vision-based AR on mobile devices [5] (using AR-Toolkit Markers) impressive progresses have been made.

To cope with the model requirements SLAM (Simultaneous Localization And Mapping) have been considered. It allows to performed the scene reconstruction and the estimation of its structure within the same framework. Since PTAM (Parallel Tracking and Mapping), which demonstrated the feasibility of a deterministic SLAM system for augmented reality on a PC [6] and on mobile devices [7], companies such as Metaio GMBH, 13th Lab or Qualcomm provide industrial and cost effective frameworks relying on computer vision algorithms. After a dedicated initialization protocol, they propose a way for the user to automatically reconstruct and track the environment and define a plane where augmented objects can be displayed. Nevertheless, such approaches lack of absolute localization and are computationally expensive in large environments.

Although it has been shown in [7] that is possible to tweak the original algorithm to use PTAM [6] on a mobile phone, its performances are greatly diminished. With the recent evolution in mobile hardware (multi-core CPU), it is now possible to run the original algorithm on such platforms, even if the provided cameras are not as good as the one used in [6]. In this paper we propose to consider a clear decoupling of the mapping and tracking step which is relevant to save computational power for the end-user application. Such approach have been successfully considered for vehicle localization [8] and augmented reality (with a known model [9]).

Furthermore, for the typical end-user, two problems emerge with such scenario. Due to the lack of absolute localization, the first issue is the difficulty to propose context aware augmentation. Although it is possible to detect known objects during the tracking [10] and display information around them, it is nearly impossible to closely register and augment the whole space of a scene, like a room or a hallway. It then becomes obvious that a complete map acquisition is required prior to decide where we want the augmentations to be placed, so that they are meaningful. The second problem lies in the ergonomic side of the application. Typically, an unbriefed end-user should not have to follow a complex protocol to be able to localize himself in the scene. The system then requires a quick and very robust re-localization in a known map without any prior information on the pose.

We here show our approach of the decoupling and explicit our first attempt to improve the end-user experience during the localization of the system in a pre-learnt and thus known map.

Fig. 1. Augmentation of a store front. On the left the live representation of the augmentations (tablet, orange phone, white phone). On the right the real environment (see also video).

2. CAMERA LOCALISATION PROCESS

The proposed system takes the form of an application suite, where three roles intervene: the map-maker, the designer and the end-user. To explicit further each role, we take the example of the augmentation of a store front (Fig. 1). One basic application idea is to allow a user/consumer to interact with a store even when it is closed.

2.1. Mapping

This is the application used by the map-maker. Its role is to produce the map, a cloud of 3D points, which will be used as a reference frame for the augmentation design and for the localization.

A two frames initialization step is required to initialize the SLAM algorithm and localize the camera relatively to the first frame. The camera is then tracked while the map-maker tries to move in the target environment to acquire as much visual features as possible. A SLAM algorithm (similar to [1]) allows the map-maker to have a visual feedback while he extends the recognizable environment for the user. Assuming a set of N observations in a sample $\{T\}$ keyframes $\mathbf{x}_i^j = (x_i^j, y_i^j, 1)^\top, i = 1..N, j \in \{T\}$, the idea is to jointly estimate the camera trajectory that is the set of pose ${}^{c_j}\mathbf{M}_W$ (where ${}^{c_j}\mathbf{M}_W$ is the homogeneous matrix that defines the camera position in the world reference frame) and the 3D structure of the scene $\mathbf{X}_i = (X_i, Y_i, Z_i, 1)^\top$ corresponding to the observation $\mathbf{x}_i$[1]. This estimation is performed by minimizing a non-linear system through bundle adjustment [11]:

$$({}^{c_j}\widehat{\mathbf{M}}_W, \widehat{\mathbf{X}}_i) = \arg \min_{{}^{c_j}\mathbf{M}_W, \mathbf{X}_i} \sum_{j \in \{T\}} \sum_{i=1..N} (\mathbf{x}_i^j - \mathbf{K}\,{}^{c_j}\mathbf{M}_W \mathbf{X}_i)^2$$

$\mathbf{K}$ being the perspective projection matrix.

During this discovery step, a set of keyframes are automatically stored with their known poses to optimize the map by bundle adjustment and to provide a set of possible references for the localization. At this point, note that the map-maker has the possibility to insert keyframes manually to further increase the coverage of reference images. The process of building on-line the 3D map in order to obtain a set of point cloud (3D points) allows a real-time feedback and qualitative assessment of the quality of the tracking and reconstruction. It also allows a better control over the success rate of the quality of the image-based localisation process.

As we will see in Section 3, at the end of this process, a more computational heavy task can occur offline, which is the computation and optimization of descriptors.

[1]For clarity purpose, we assume a constant of observed feature over the image sequence, which is in practice obviously not true.

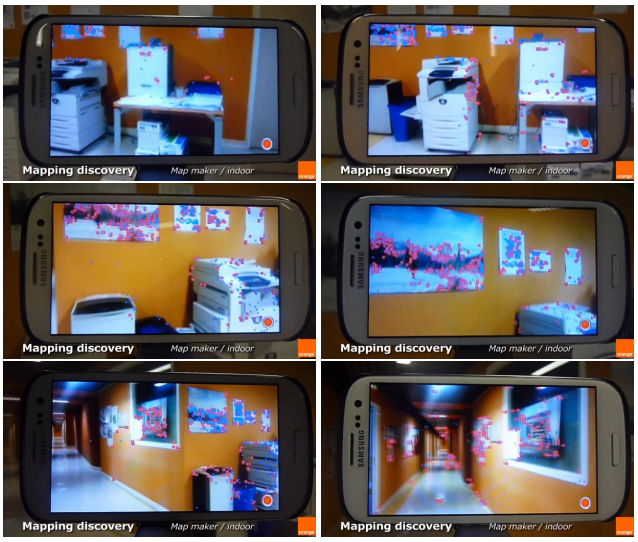

Fig. 2. Tacked keypoint for scene reconstruction through bundle adjustment (red keypoint are the reconstructed keypoints).

2.2. Augmentation Design

The augmentation designer has to place each augmentation (virtual objects) in the reference frame of the whole map. This allows for augmentations anywhere in the scene where a part of the map can be observed and tracked, which is more comfortable than just around a set of fixed patches.

To handle occlusions and real object interactions, a 3D model of the scene can be acquired by a multiple views structure from motion, and is registered with the point cloud produced in the Mapping step. From there, the task of the designer is quite easy. He has to put the virtual object anywhere he wants in regard of the textured 3D model of the scene. This step can be done offline with a modeling tool (Unity) or online using the tracker to place augmentations directly in the real environment with the mobile phone.

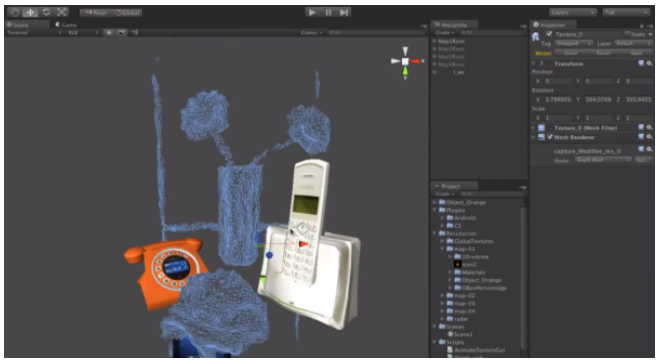

Fig. 3. Off-line augmentation designer.

2.3. Localization

This app is used by the end-user. It tracks the map, re-localizes itself (pose computation), from the geolocalized stored keyframes (a by-product of the SLAM algorithm). As in the mapping process, the goal is to localize the camera, that is estimating the pose ${}^{c}\mathbf{M}_{W}$ using observation $\mathbf{x}_i, i = 1..N$ and the 3D structure of the scene $\mathbf{X}_i, i = 1..N$ (3D point cloud) that has been provided by the mapping process. In that case pose can be computed using algorithm such as POSIT [12] or non-linear poses estimation algorithm [13] which aimed at estimating for the current pose ${}^{c}\mathbf{M}_{W}$ the forward projection error given by:

$$ {}^{c_j}\widehat{\mathbf{M}_W} = \arg\min_{{}^{c_j}\mathbf{M}_W} \sum_{i=1..N} (\mathbf{x}_i - \mathbf{K}{}^{c}\mathbf{M}_W\mathbf{X}_i)^2 $$

In this part, all map discovery and optimization features (such as bundle adjustment) are disabled, reducing the amount of computation needed.

Exceptionally, the system can still behave as a real SLAM system for short periods of time to allow the user to move a little bit outside of the designed environment but it is not intended to create a large map and forgets rapidly new features.

3. IMPROVING RE-LOCALIZATION

During previous experiments, it was established that the re-localization of the end-user from an unknown pose, should work anywhere and from any point of view in the augmented scene, in less than a few seconds. The goal is clearly to register 2D features with 3D key point. During the mapping step, a descriptor (FREAK [14]) have been attached to each keypoint [15].

If we look at the real space where the user would like to be able to locate itself, we can roughly evaluate the performance of the proposed re-localization algorithm, taking only a few significant orientations of the camera, see Fig. 4.

We first consider an image-based re-localization where we use the position of a keyframe (used in the mapping process),

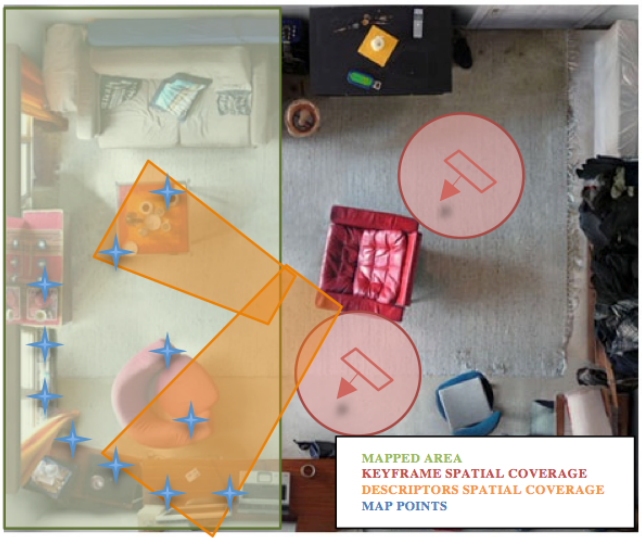

Fig. 4. An illustration of a 2d spatial re-localization coverage where the user can move in the whole room.

as the starting pose of the camera for the tracker. It is done by a ZMSSD on a subsample of the real image, with their orientation aligned along the Z axis. This method allows for real-time re-localization but has a pretty small spatial coverage around the reference keyframe. We can clearly see that although we have the capability to manually add keyframes, it is not a viable solution for a large 3D scene.

To increase the coverage, we decided to use FREAK descriptors on each map keypoint and in each keyframe during the mapping step. They are then matched with descriptors computed on the current frame(FAST corners), which have a Shi-Tomasi score beyond a threshold on the two lowest levels of the image pyramid. A RANSAC-based POSIT [12] then determines the new pose of the tracker as described in section 2.3.

Although it is efficient for frame to frame matching the first method is fast but quite unreliable for re-localization issue, while the second is more robust but significantly slower. We decided to use both at the same time, running them concurrently in two threads. We chose a strategy consisting of using the image-based re-localization while the descriptor-based is still running. If we are still lost when the second one finishes, we use its result instead.

4. RESULTS

The system has been tested on both Android and iOS. The mobile phones were Samsung Galaxy S2, Samsung Galaxy S3 and iPhone 4S. On both the Galaxy S3 and the iPhone 4S (which are not current high-end mobile phone) the tracking and image-based re-localization are done in real-time for an image of sizes, respectively 320x240 and 480x360.

A descriptor based re-localization lasts between 800ms and 1500ms when it's a success. The framerate on the Samsung GS2 is near real-time, and can only really be used with the image-based re-localization. The descriptor-based re-localization increases the spatial coverage by a factor three. For all the results shown in this section, a maping process using the bundle adjustment framework has been previously done.

Fig. 5 shows an experiment carried out on a commercial showcase with a limited workspace. In that case re-localisation is not a difficult issue. Note that, occlusions of virtual objects by real one are handled thanks to a complete reconstruction of the scene using a Kinect (see Figure 3).

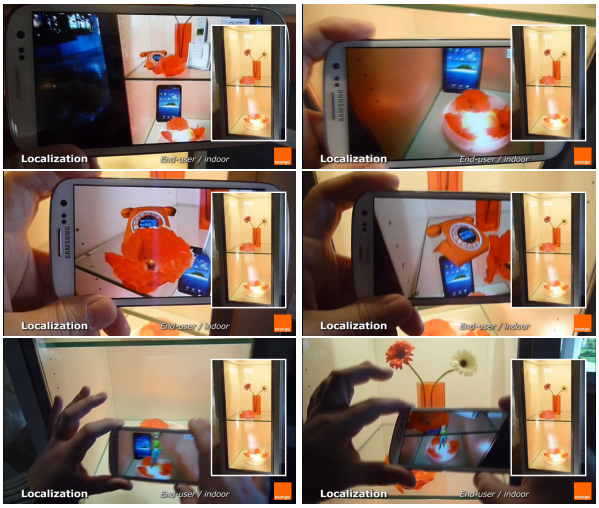

Fig. 5. System in action in a showcase

Fig. 6 shows the localization process for an outdoor scene (image acquired at night) while Figure 7 shows a virtual discussion scene inside a large atrium. Top of Fig. 7 shows the keypoint tracked and registered with the 3D models to compute the actual camera pose.

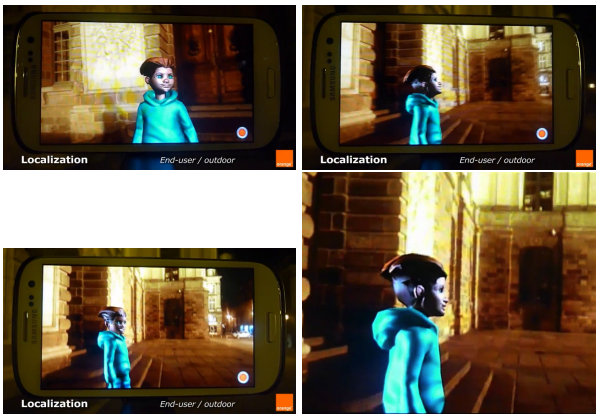

Fig. 6. System in action for an oudoor scene at night

Fig. 7. System in action an indoor scene (top image shows the keypoints used in the relocalisation process).

Video: a video of the system is visible at this address `http://youtu.be/ibEsHg2k1yQ`

5. CONCLUSION AND FUTURE WORKS

We have established various benefits of decoupling localization and mapping for augmented reality. It is meaningful for performance and optimization, and is mandatory when we want to augment contextually the environment. For future work, we plan to do further optimizations on the mapping step to allow for a better re-localization and a quicker tracking, such as making statistics on the value of the information contained in each version of a map point descriptor.

6. REFERENCES

[1] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *IEEE Trans. on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, July 2006.

[2] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3d tracking using online and offline information," *IEEE Trans. on PAMI*, vol. 26, no. 10, pp. 1385–1391, October 2004.

[3] G. Simon and M.-O. Berger, "Registration with a zoom lens camera for augmented reality applications," in *ACM/IEEE Int. Workshop on Augmented Reality, IWAR'99*, San Francisco, CA, Oct. 1999, pp. 103–114.

[4] M. Pressigout and E. Marchand, "Model-free augmented reality by virtual visual servoing," in *IAPR Int. Conf. on Pattern Recognition, ICPR'04*, Cambridge, UK, Aug. 2004, vol. 2, pp. 887–891.

[5] D. Wagner and D. Schmalstieg, "First steps towards handheld augmented reality.," in *Int. Conf. Wearable Computers, ISWC'03*, 2003, vol. 3, pp. 127–135.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[7] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*, Orlando, October 2009.

[8] E. Royer, M. Lhuillier, M. Dhome, and J.M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 237–260, 2007.

[9] M. Tamaazousti, V. Gay-Bellile, S. Collette, S. Bourgeois, and M. Dhome, "Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment," in *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2011*, 2011, pp. 3073–3080.

[10] C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *IEEE Int. Conf. on Robotics and Automation, ICRA'03.*, 2003, vol. 1, pp. 842–849.

[11] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment: A modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds., vol. 1883 of *Lecture Notes in Computer Science*, pp. 298–372. Springer Berlin Heidelberg, 2000.

[12] D. Dementhon and L. Davis, "Model-based object pose in 25 lines of codes," *Int. J. of Computer Vision*, vol. 15, pp. 123–141, 1995.

[13] E. Marchand and F. Chaumette, "Virtual visual servoing: a framework for real-time augmented reality," in *EUROGRAPHICS'02 Conf. Proceeding*, G. Drettakis and H.-P. Seidel, Eds., Saarebrücken, Germany, Sept. 2002, vol. 21(3) of *Computer Graphics Forum*, pp. 289–298.

[14] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'12*, 2012, pp. 510–517.

[15] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *IEEE Int. Conf. on Computer Vision*, Beijing, China, 2005, vol. 2, pp. 1508–1515.