Augmented Reality in Large Environments: Application to Aided Navigation in Urban Context

V. Gay-Bellile* CEA, LIST
P. Lothe CEA, LIST
S. Bourgeois CEA, LIST
E. Royer LASMEA (CNRS / UBP)
S. Naudet Collette CEA, LIST

Abstract

This paper addresses the challenging issue of vision-based localization in urban context. It briefly describes our contributions in large environments modeling and accurate camera localization. The efficiency of the resulting system is illustrated through Augmented Reality results on large trajectory of several hundred meters.

Index Terms: H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities—Life Cycle; I.4.8 [Scene Analysis]: Object Recognition—Tracking

1 Introduction

This paper describes a vision based system for aided navigation in large urban environments. We use a single pinhole camera mounted on a vehicle. The goal is to insert on a screen virtual useful information in real-time to guide the driver in large cities. The main challenging problem is to accurately localize the camera during a long road trip.

Vision based localization in large environments usually works in two steps: off-line environment modeling and then on-line localization in this map, see *e.g.* [1]. By contrast, in small (or multiple small) environments, those two steps are both done on-line, see *e.g.* [2]. This paper addresses both large environments modeling and accurate localization in urban context. It aggregates our different works in a complete system for Augmented Reality purposes.

We initially summarize our framework for automatically building and georeferencing drift free map of large urban environments. The method we propose relies on a coarse 3D city model to correct the drift of Structure-from-Motion point cloud. Then, we introduce a new real-time hybrid localization algorithm that combines Viewpoint Recognition and on-line Structure-From-Motion. Finally, we present Augmented Reality results on large trajectories of several hundred meters.

2 Off-Line Environment Modeling

This section describes a process which aims to create a 3D landmark database of a city center from a single pinhole camera mounted on a car. Structure-from-Motion is usually used to tackle this problem. For example, a full process is proposed in [3] to create a 3D point cloud from an image set and georeference it by using a satellite image. The reconstructed model is assumed drift free. It is almost true in their application scope since a multitude of different viewpoints are available.

In our context, *i.e.* a driver navigating in a city, the drift of SfM is unavoidable since the geometrical constraints are weak: a 3D point is observed in few consecutive images with almost similar viewpoints. Consequently, the resulting reconstruction can be far from the real geometry of the scene and thus can not be georeferenced with the algorithm described in [3].

We propose an alternative solution that simultaneously corrects *a posteriori* the reconstruction drift and georeferences the resulting point cloud. Our main idea consists of using the geometric constrains provided by a coarse 3D city model, *i.e.* a set of textureless planes (see *e.g.* figure 1(a)), to improve the accuracy of SfM reconstructions. City models tend to be widely spread (*e.g.* Google Earth) and are globally consistent, *i.e.* the 3D information they provide is drift free. Furthermore, their precision is almost satisfactory (≈ 1 meter).

The *a posteriori* drift correction module is composed of two subprocesses (more details are available in [4]):

- First, the estimated trajectory is segmented through a classical polygonalisation process and an ICP-like algorithm is then used to estimate the piecewise similarity that optimally fits the 3D point cloud onto the city model. Here, the goal is to correct the large reconstruction deformations and to regain its global consistency. Nevertheless, at this step, the reconstruction is not accurate enough for Augmented Reality purposes.

- Secondly, a bundle adjustment step is used to correct the residual local error of the reconstruction. In order to keep the constraints brought by the 3D city model, a specific cost-function has been designed. It includes both image information and city model constraints in a single term. This cost-function is a reprojection error which encourages optical rays to converge onto the city model.

3 On-line localization: Combining Viewpoint Recognition and Structure-From-Motion

Standard localization processes, *e.g.* [1], use a Viewpoint Recognition (VR) algorithm (based on vocabulary tree in our case) to associate 3D features of the database with interest points extracted from the current images. The associated pose can then be computed. This process is sensitive to viewpoint and lighting variations between the database and the current view. Moreover, recognition ambiguities are common in urban context since many buildings may share the same appearance. Therefore, we introduce a new localization process that combines VR and SfM algorithms. It presents the advantages to be more robust to viewpoint variations and lighting conditions, and provides a pose in unmodeled parts of the environment. We also briefly describe tools to handle ambiguities in viewpoint recognition.

The fusion process.

One possibility is to combine VR and SfM by sharing the same map. The SfM algorithm is then used to add 3D features in the database that reflect the current lighting conditions and viewpoints. The pose is then estimated more robustly. This fusion process assumes that a sufficient number of 3D features of the database are continuously matched to prevent from SfM drift. Otherwise, the map will be corrupted by erroneous 3D features. It is well adapted in many scenarios, *e.g.* small indoor AR workspaces. However, balancing 3D points with different origins (off-line database vs on-line SfM) is much more complicated in large urban context: SfM drift is unavoidable and the number of

*mail: CEA, LIST, Vision and Content Engineering Laboratory, Point Courrier 94, Gif-sur-Yvette, F-91191 France e-mail: vincent.gay-bellile@cea.fr recognized features is not always high mainly due to lighting condition instabilities in outdoor environments. Then, we propose an alternative solution that copes with the specificities of the targeted application.

Our data fusion scheme uses VR to initialize the SfM and punctually correct its error accumulation. This correction only occurs when a SfM-keyframe is successfully matched (through Viewpoint Recognition) to the database and if the associated pose is estimated with a good confidence (measured by the ratio of inliers vs outliers in RANSAC and the dissemination of the inliers image observations). In this case, current SfM errors in position and orientation are given by the difference between the poses returned by the VR and the SfM algorithms whereas the dilatation / contraction error is measured as the ratio of the two trajectories length or by comparing the features common to both maps. The drift correction module estimates the similarity that, once applied to the SfM map, minimizes the errors described above. This transformation is then used to change the position, orientation and scale of the SfM map.

Handling ambiguous viewpoints. Localization based on recognition through vocabulary tree is subject to ambiguous viewpoint since it encodes all the descriptors of the database. This is especially true in urban environments since different building fronts may have similar appearance. To reduce such ambiguities, we take benefits from the pose returned by the SfM algorithm to limit the descriptors space to those associated with the 3D points observed by the nearest cameras in the database. An exhaustive matching is then quick enough. The tree structure is not used. However, when recognition fails for a long time interval, the poses returned by the SfM algorithm are not certain enough. Thus, in this case, Vocabulary Tree recognition has to be used. Ambiguities are then tackled by conserving the most likely localization hypotheses returned by the Vocabulary Tree that are checked with the following frames.

4 Experimental Results

4.1 Building the database

We use a low-cost IEEE1394 GUPPY camera providing gray-level images with 640x480 resolution at 30 frames per second. Two video sequences have been acquired along 1500 and 2000 meters trips in Versailles, France. Without correction, the SfM reconstructions[1] are far from the real geometry of the scene: in the end, errors in position are up to 70 and 150 meters respectively. After our process, this drift is corrected along all the trips. We superimpose the final reconstructions on a satellite image as illustrated on figure 1. It shows that the camera trajectories follow the road between the buildings. The reconstructed point clouds also regain their consistency and are accurately registered on the building contours. The remaining errors mainly come from inaccuracies in the used 3D city model. The two final reconstructions form a single georeferenced database of more than thirty thousand features. Their associated descriptors are then encoded in a vocabulary tree structure.

4.2 Real-time localization

Two others sequences have been acquired. The two trips (represented in orange and magenta in figure 2) are both 650 meters long. They cross parts of the environment modeled with the two learning sequences. An unmodeled street is also traveled in the second sequence. The localization process described in §3 provides camera poses with a sufficient accuracy for Augmented Reality, even in unmodeled and unrecognized part of the city. Figure 2 illustrates Augmented Reality application through the insertion of virtual navigation information.

Our localization process is highly real-time. SfM and VR are performed on two different cores in parallel. The mean frame rate is approximately 40 fps on a Pentium IV dual-core 3GHz.

[1] We use the key-frames SfM algorithm described in [5].

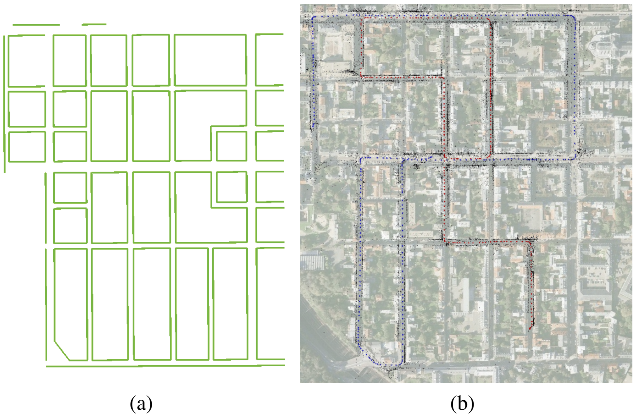

Figure 1: (a) Top view of the coarse 3D city model used in our experiments. (b) The final database superimposed on the satellite image. It comes from two georeferenced SfM point clouds. Their drift has been corrected with the process described in §2.

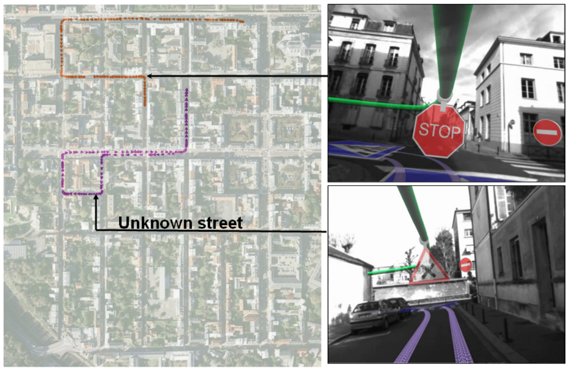

Figure 2: Left: the two trajectories used to evaluate the localization algorithm described in §3. Right: Augmented Reality application through insertion of virtual navigation information.

5 Conclusion

We present in this paper a complete vision-based system for Augmented Reality in large urban environments. We contribute to both large environments modeling and camera localization. Experimental results illustrate the accuracy of the proposed system through aided navigation scenarios. Further work will devote to on-line SfM drift correction by the aid of coarse 3D city models.

References

[1] C. Arth, D. Wagner, M. Klopschitz, and A. Irscharai. Wide area localization on mobile phones. In *ISMAR*, 2009.

[2] R. Castle, G. Klein, and D. Murray. Video-rate localization inmultiple maps for wearable augmented reality. In *ISWC*, 2008.

[3] R. Kaminsky, N. Snavely, S. Seitz, and R. Szeliski. Alignment of 3d point clouds to overhead images. In *CVPRW*, 2009.

[4] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Towards geographical referencing of monocular slam reconstruction using 3d city models: Application to real-time accurate vision-based localization. In *CVPR*, 2009.

[5] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyse, and P. Sayd. Real-time localization and 3d reconstruction. In *CVPR*, 2006.