MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

Augmenting off-the-shelf paper maps using intersection detection and Geographical Information Systems

Liming Yang
Ecole Centrale de Nantes - CRENAU
Nantes, France
liming.yang@ec-nantes.fr

Jean-Marie Normand
Ecole Centrale de Nantes - CRENAU
Nantes, France
jean-marie.normand@ec-nantes.fr

Guillaume Moreau
Ecole Centrale de Nantes - CRENAU
Nantes, France
guillaume.moreau@ec-nantes.fr

Abstract

This paper presents our recent research on augmented maps where unprepared paper maps are used and Geographic Information System serves both as reference database and augmentation data. We use road intersections to register paper map and GIS so several paper maps of the same city whatever their styles and colors can be augmented using the same reference database. We present a new road extraction method to get better road intersection for a quick initialization and a faster method to register maps and GIS. Different maps are used to demonstrate the effectiveness of our method.

1 Introduction

Augmented maps have been a popular research topic in the augmented reality community for several years. They present the advantages of both paper maps, which are easier to intuitively manipulate, facilitate people's communication, and those of electronic maps, which are dynamic thus be able to offer different contents according to different audiences.

Geographic Information System (GIS) is the ground layer for electronic maps. Most information of cities is now recorded in such a system. It is a natural way to base augmented maps on GIS as well. Commonly, there are three key steps for augmenting a map: feature extraction, GIS-registration and tracking. The feature extraction stage depends on the GIS-registration methods being used. Registration methods can be grouped in two categories depending on whether or not they rely on fiducial markers.

Fiducial markers can be intrusive for users and require map preparation before tracking can be achieved. Many researchers aimed at removing or reducing this intrusiveness. Reilly et al. [6] attached RFID tags with different IDs behind a paper map and a special RFID reader is required on hand-held devices. Schöning et al. [8] used ARToolkitPlus markers on maps and rendered the marker with electronic maps from a video see through device. Point markers are also used for registration on streets intersections [11] or on landmarks [4].

Besides fiducial markers, natural features have also been used on augmented maps such as SIFT [5] or modified SIFT descriptors [7]. However, these texture-based methods are map dependent. They can only register one kind of maps whereas city maps can be represented using many different colors or textures. This represents the main limitation of such approaches.

The robust random dot markers (RRDM) [12] proposed by Yang et al. allows for registering maps with a GIS-database using only pure geometric features. They extracted road intersections from paper maps, calculated intersections in GIS, and matched these two point patterns. As analyzed by AlHalawani et al. [1], this approach is map independent.

In this approach, road intersection extraction is crucial. False road intersections can introduce a poor performance of the method, or even a total failure. A usual way to find road intersections on a map is to identify the road network, extract the skeleton before finding lines intersections. The difficulty lies in the treatment of texts on maps, such as road names, which severely reduce the quality of the automatically extracted skeleton. Callier et al. [2] used local line probability to find road pixels and used color histograms to find road layers. Chiang et al. [3] used Mean-shift, Median-cut and K-means to find road layers and text/graph separation techniques are used. But none of these methods is efficient enough to allow for real-time augmentation. In [12], Yang et al. used a modified version of the above methods but a map-dependent filter has to be used to speed-up the process.

Our overall method is similar to the one presented in [12], but we use a different method for road intersection extraction which is more efficient and does not require the use of map-dependent filters. A general framework for augmenting paper maps is also presented in order to achieve real-time tracking.

2 Proposal

Our framework has three modules: road intersection detection, map-GIS registration and tracking. Reference intersections coordinates are calculated and cleaned beforehand since there are small roads which will never present on a city map. Map intersection are detected from real maps by our road intersection detection module in the very first frame during the initialization (cf. Section 2.1). Then map intersection and reference intersection are matched by the map-GIS registration module using a modified RRDM method also during initialization (cf. Section 2.2). At last, Section 2.3 explains how to integrate a traditional SURF tracking method into the framework for real-time tracking.

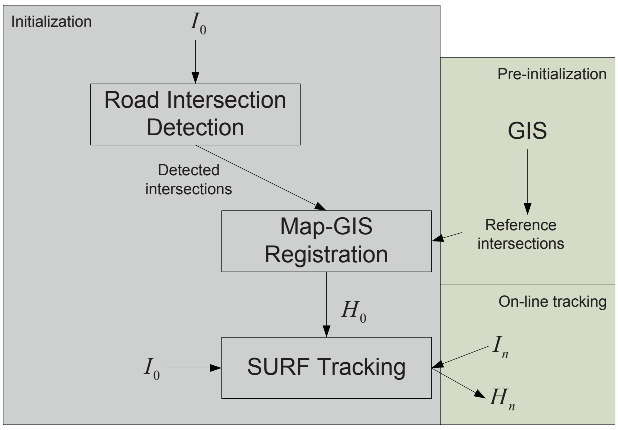

Figure 1. Schema of the application. I_0 is the first image for initialization. H_0 is homography between GIS and the map in I_0

2.1 Intersection detection on real maps

For road intersection detection, we firstly use a modified level set method to extract road layers. Then, we skeletonize the road layer, extract junctions and clean junctions that are too near to each other. Our contribution is mainly in road layers extraction improvement, which is shown in Fig.3. At first, the user picks a single road pixel from the image (a seed point). This is very easy to do with a mouse or on a tactile screen. From this pixel, road color and width can be estimated. Then, a classic level set method is used to find the map region to avoid false detections coming from out of the region of interest. At last, a modified level set method is applied to find road layers inside the map region.

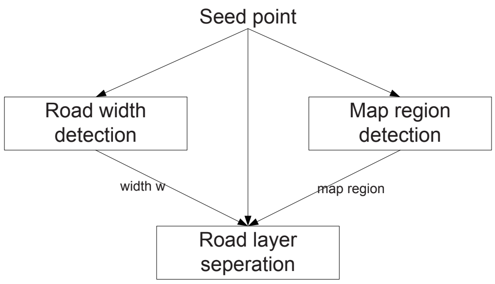

Figure 2. Workflow of road layer extraction

We use a 50×50 local image portion centered at the seed point to find the road width since road width on a map is usually far less than 50 pixels. A Canny edge detector is used to extract road edges and a flood fill at the seed point is applied to find local road layers, which containing R road pixels. We then dilate the local road layer by 1 pixel and the result contains Q road pixels. Since roads can be modeled as thin rectangles, road width w can be easily calculated as:

$$w = \frac{w_1 + w_2}{2} \quad \text{with } w_1 = 2R/(Q-R), w_2 = 2R/(Q-R+4) \tag{1}$$

where w_1 is for the case where a single road is present in the local image portion while w_2 is for the case where a crossing is present. These are two most common cases in a local map.

The general idea of level set method is to minimize $E(\Gamma)$ in eqn.2 where Γ is the contour which separates segmentations A_1 and A_2. $\mathbf{u}(x, y)$ is the color at (x, y). $\mathbf{C_1}$ and $\mathbf{C_2}$ are expected colors for two segmentations respectively. Shi's implementation is used as our classic level set method[9].

$$E(\Gamma) = \int_{A_1} |\mathbf{u}(x,y) - \mathbf{C_1}| + \int_{A_2} |\mathbf{u}(x,y) - \mathbf{C_2}| \tag{2}$$

Once we get the map region, a modified level set method is applied since classic level set doesn't work well for road extraction. The classic level set uses a rectangle contour which is just a little smaller than the target image as initial contour. It can segment "simply connected space" but can not identify holes in that space since the contour cannot appear from nowhere. But road networks are not a simply connected space, they divide a map into many unconnected areas, or "holes" (see in Fig.3 top right). Moreover, the resulting contours usually lie on pixels with high color gradients. So texts on the map can often prevent the contour from moving thus result in poor segmentation results (see in Fig.3 bottom left).

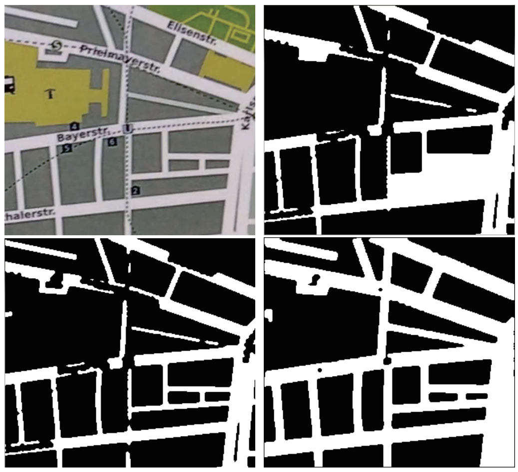

Figure 3. Results of various level sets. Top left: original map portion. Top right: classic level set method. Bottom left: texts disturb contours. Bottom right: our modified level set.

To overcome the first problem, we initialize the level set method with rectangular stripes (cf. Fig.4). Each stripe has a width of $2w$. This ensures that each "hole" is divided by initial contours. For the second problem, since thickness of texts is usually very small, we apply an operation in Eqn.3 to each pixel to remove them. Moreover, C_1 in Eqn.2-3 is set to the color of the seed point. C_2 is set to the average value of pixels in A_2.

$$\mathbf{u}(x,y) = \operatorname*{argmin}_{|x'-x|\le 2, |y'-y|\le 2} |\mathbf{u}(x',y') - C_1| \tag{3}$$

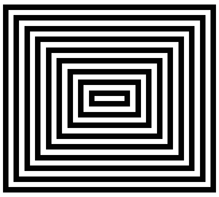

Figure 4. Example of rectangular stripes. Initial contours are located between white stripes and black stripes.

2.2 Map-GIS registration

We use RRDM [12] for map-GIS registration. It is a robust method for matching coplanar points under perspective transformations. Authors have shown its ability to match detected road intersections even with an important detection noise. The method relies on geometric local distribution of points to characterize each point. For each interest point, Quad-Points and affine invariant TSR descriptors are constructed from the interest point itself along with its k-nearest neighbors. These TSR descriptors serve as "geometric labels" of the interest point. With the help of these labels, a local voting process is employed to establish a pre-alignment between reference intersections and detected intersections. Then, coherency between local affinities is used as a criteria to find inliers and give a gross result. At last, a recovery mechanism is used to improve the result.

In the original approach of [12], local votes had to be processed and local affinity had to be estimated for each possible reference-detected intersection correspondence before finding the final result. We found that this work was somehow redundant and chose to use a RANSAC-like approach. In this way, local voting and local affinity are calculated only when it was necessary. This has significantly improved the matching speed with no impact on robustness (see Fig.5).

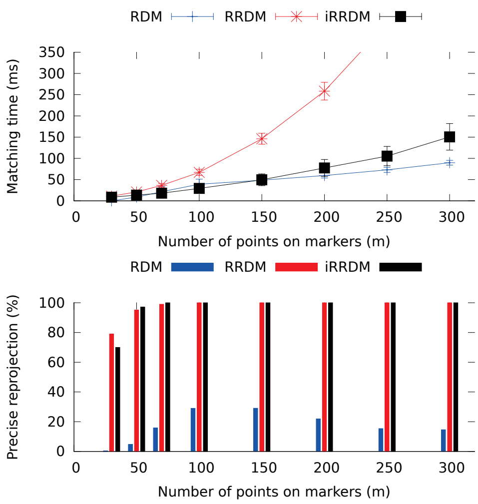

Figure 5. Efficiency of improved RRDM (iRRDM) against RRDM [12] and RDM [10].

2.3 Tracking

Since road intersection extraction contains many image manipulations and RRDM requires hundreds of milliseconds per matching, a real-time tracking can hardly be achieved by using this combination. We choose to use traditional feature point method which is robust and suitable for 2D textured planar objects in the tracking module. We use the SURF implementation of OpenCV.

During the initialization step, we create the library of model features using the first image I_0. The estimation of homography H_0 (cf. Section 2.2) is also fed to the module. During the online tracking phase, SURF features are extracted from each image I_n and matched with the model features to give an incremental homography ΔH_n. Then, the final homography H_n between the GIS and the map in image I_n can be calculated as:

$$H_n = \Delta H_n H_0 \qquad (4)$$

To improve the stability of tracking, every 20 frames, we renew the model feature library with current frame and update H_0.

3 Results

We recorded three videos of different paper maps of Munich with an iPad-air camera to test our method. We use OpenStreetMap data as GIS database and raster maps found on the Internet and printed(see Fig.6).

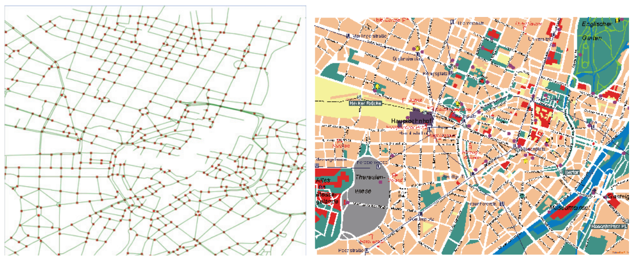

Figure 6. Data resources. Left: GIS road network (green) and road intersections (red). Right: One raster map.

We firstly compare results of intersection detection using filter-based method and our modified level set method in Fig. 7. Detection results are greatly improved by using our method while filter-based method has many false detections especially near texts.

Initialization results for two videos are presented in Fig. 7. GIS roads are placed exactly onto roads of paper maps. Several tracking frames of the video are in Fig. 9. We can see that the position of augmented information remains correct for most of time. But small drifts appear due to SURF tracking at the end. As to time performance, although registration takes 500∼1000ms, tracking works at interactive rate, i.e. 20∼30Hz.

4 Conclusion and future works

We presented a framework for augmenting unprepared paper maps with GIS database by matching

Figure 7. Intersection detection results (Blue points). Top: filter-based method of [12]. Bottom: Ours.

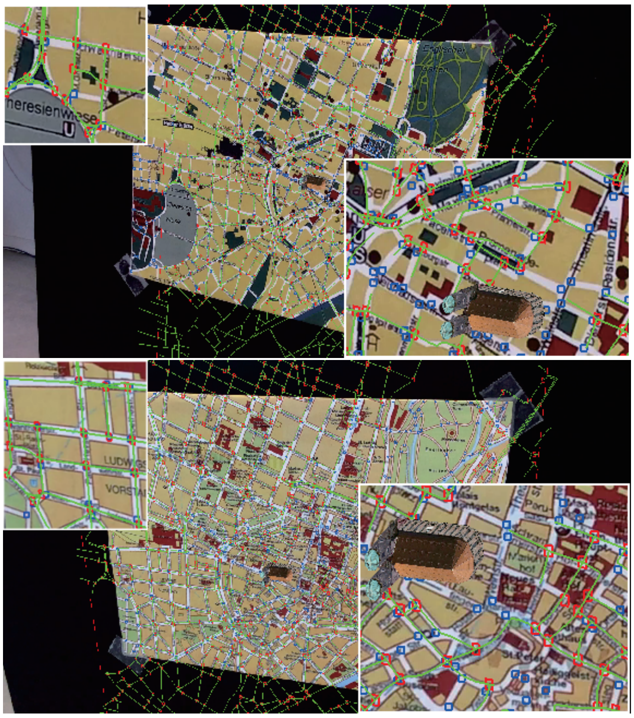

Figure 8. Initialization results for two different maps.

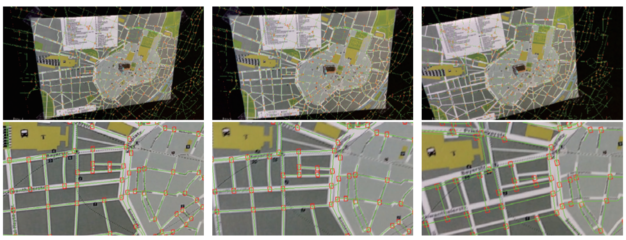

Figure 9. Frame 11, 94 and 167 (from left to right) of tracking results in video 3. Top row presents original augmented frames while bottom row shows zooms to illustrate the precision of the matching.

road intersections. To speed up the initialization step, a modified level set method is introduced to extract the road layer from a paper map, which is robust against overlaps between texts and roads. An improved RRDM method is also used to speed up registration. Three videos of three different maps are augmented to show the efficiency of the method.

We plan on introducing GPU calculation for processing the modified level set method. This is the most time-consuming part and can be done in parallel. Augmenting maps of different scales for the same city is also one of our interests.

References

[1] S. AlHalawani, Y.-L. Yang, P. Wonka, and N. J. Mitra. What makes London work like London? *Computer Graphics Forum*, 33(5):157–165, 2014.

[2] S. Callier and H. Saito. Automatic road area extraction from printed maps based on linear feature detection. *IEICE Transactions*, pages 1758–1765, 2012.

[3] Y.-Y. Chiang and C. Knoblock. A general approach for extracting road vector data from raster maps. *International Journal on Document Analysis and Recognition (IJDAR)*, 16(1):55–81, Oct. 2011.

[4] S. Martedi and H. Saito. Augmented Fly-through using Shared Geographical Data. In *21st International Conference on Artificial Reality and Telexistence*, ICAT 2011, pages 47–52, Osaka, Japan, 2011.

[5] A. Morrison, A. Mulloni, S. Lemmelä, A. Oulasvirta, G. Jacucci, P. Peltonen, D. Schmalstieg, and H. Regenbrecht. Mobile Augmented Reality: Collaborative Use of Mobile Augmented Reality with Paper Maps. *Comput. Graph.*, 35(4):789–799, 2011.

[6] D. Reilly, M. Rodgers, R. Argue, M. Nunes, and K. Inkpen. Marked-up Maps: Combining Paper Maps and Electronic Information Resources. *Personal Ubiquitous Comput.*, 10(4):215–226, Mar. 2006.

[7] G. Reitmayr, E. Eade, and T. Drummond. Localisation and Interaction for Augmented Maps. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 120–129, Washington, DC, USA, 2005. IEEE Computer Society.

[8] J. Schöning, A. Krüger, and H. Müller. Interaction of mobile camera devices with physical maps. In *Adjunct Proceeding of the Fourth International Conference on Pervasive Computing*, pages 121–124, 2006.

[9] Y. Shi and W. C. Karl. A real-time algorithm for the approximation of level-set-based curve evolution. *Image Processing, IEEE Transactions on*, 17(5):645–656, 2008.

[10] H. Uchiyama and H. Saito. Random Dot Markers. In *Proceedings of the 2011 IEEE Virtual Reality Conference*, VR '11, pages 271–272, Washington, DC, USA, Mar. 2011. IEEE Computer Society.

[11] H. Uchiyama, H. Saito, M. Servières, and G. Moreau. Camera tracking by online learning of keypoint arrangements using LLAH in augmented reality applications. *Virtual Reality*, 15(2-3):109–117, 2011.

[12] L. Yang, J.-M. Normand, and G. Moreau. Robust random dot markers: towards augmented unprepared maps with pure geographic features. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 45–54. ACM, 2014.