Augmented fly-through using shared geographical data

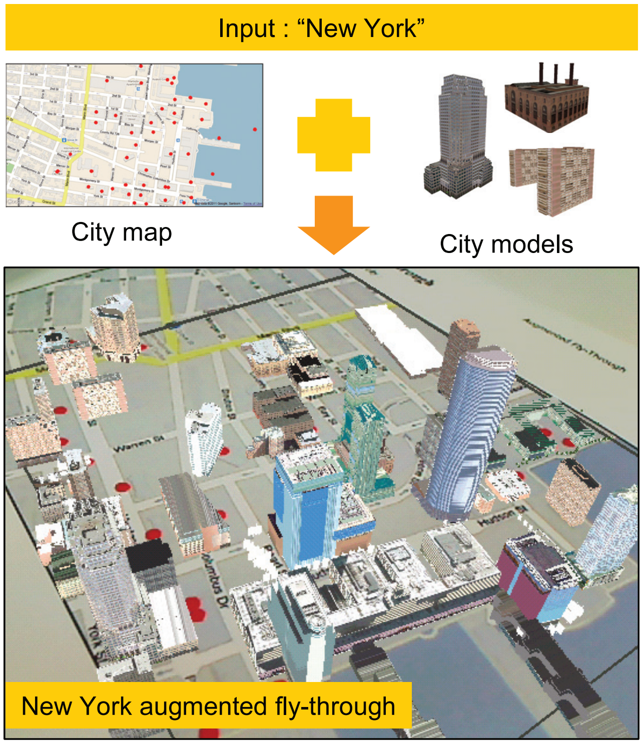
Sandy Martedi*
KEIO University

Hideo Saito†
KEIO University

Abstract

This paper presents a development of augmented maps using shared geographical data. In general, augmented reality applications use predefined 3D models data. In our collaborative approach, we utilize shared geographical data from Google maps and city models from Google 3D Warehouse. Our system allows users to share and use maps and city models of any location in the world. The user then can print the maps and view the overlaid virtual city models through a web camera or head mounted display (HMD). We explore suitable tracking methods on three types of maps: normal (default), terrain and and satellite maps. Finally, we present an augmented fly-through application where the user can browse and view the 3D models on a paper map.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities;



©2011 Google - Map Data ©2011 Google, Sanborn

Figure 1: System overview. First, the user inputs a city name in our data extraction tool, for instance "New York". The user then downloads the 3D models from the 3D warehouse. The user prints the map and views the overlaid 3D models on the map.

1 Introduction

Some approaches and applications for visualizing the digital geographic data have been explored intensively. For example, geographical information system (GIS) accommodates the creation, management and visualization of geographic data such as digital maps and 3D city models. Generally, GIS data contain layers and localized data including numbers, vectors, images, and text. Existing GIS is used only for overlaying those 2D information on digital maps. State of the arts of GIS tried to enhance the visualization of maps by adding more informations including multimedia and 3D models in order to support geographical data analysis. For instance, Bing Map from Microsoft attempted to visualize GIS in form of surveillance and satellite maps in real time [3]. On the other hand, Google map, earth and 3D warehouse are developed to visualize geographical data seamlessly on desktop computers and mobile devices [5]. Especially, 3D warehouse supports the creation of city models and collaborative 3D modeling using geographic coordinate as the shared space.

To enhance 3D GIS visualization, augmented-reality-based systems have been developed. A system so called augmented maps combined digital layers and a paper map. Augmented maps overlays virtual data such as city names, region descriptions or 3D models of landmarks and buildings on top of a paper map.

In general, 3D models are prepared beforehand and dedicated only for a specific augmented maps application. It is feasible only for a particular region. However, to realize augmented maps for any location, preparing many maps and 3D models can be time-consuming.

To solve this issue, we are interested in the collaborative solution by sharing geographical data that available on the server such as Google Maps and 3D Warehouse. This solution allows users to download and use the shared data for their augmented maps application. Particularly, since Google maps and 3D warehouse lay on the same geographic coordinate, it is possible to merge or augment them straightforwardly.

We aware that the main difficulty of augmented maps is how to detect and track the paper map. Therefore, we need to study a suitable detection and tracking method for a paper map.

Image analysis for image tracking such as map indexing is feasible by separating the map into some layers such as roads, intersections or regions. It becomes easier to extract important features of map from those layers for detection and tracking. However, usually those layers are not accessible on a map server such as Google Map. Instead, only points data added by users and the map image are usually available. Therefore, additional image analysis and data preparations are necessary to create a track-able map for augmented maps.

We propose an approach on preparing a map for augmented maps by adding a tracking layer above the background map instead of separating layers from the map image. This approach gives us a freedom to choose any features for tracking regardless the type of maps we will use. Therefore, this approach is applicable for any type of map. We also develop a tool for retrieving the trackable maps and 3D city models from Google Maps and 3D Warehouse.

*e-mail: sandy@hvrl.ics.keio.ac.jp
†saito@hvrl.ics.keio.ac.jp

The 21st International Conference on Artificial Reality and Telexistence
November 28-30, 2011, Osaka, Japan
ISSN: 1345-1278 © 2011 The Virtual Reality Society of Japan

The user can download the desired maps and virtual contents by inputing a city name. The next step is printing the map and prepare a web camera or HMD. Finally, the user can view the 3D city models are overlaid on top of the paper map.

Our system improves the prior approaches on augmented maps. We also insist that our system is a novel system that combines the augmented reality with shared model data available on the Internet. To the best of our knowledge, there is no other research that has explored the augmentation of shared 3D city models by adding tracking layer above the background map. Finally, we demonstrate our approach by developing an augmented fly-through application that allows user to explore augmented maps using the web camera as a pointer.

The rest of the paper is organized as follows. The prior researches are described in the Sec. 2. We explain our proposed work in the Sec. 3 followed by the description of our augmented fly-through application in Sec. 4. We evaluate our scheme using some tracking methods in terms of the number of matches and computational cost in Sec. 5 and discuss it in Sec. 6. Finally, we conclude this paper in Sec. 7.

2 Related Works

AR toolkit has been introduced in the early stage of augmented reality [13]. AR toolkit is initially used for a collaborative conference system. There have been some researches on utilizing AR toolkit for developing augmented maps. Hedley et al. combined the augmented reality with geographical data visualization [11]. They also equipped the system with fingertip detection and interaction. Bobrich et al. also used the AR toolkit to track a planar map [7]. McGee at al. developed a collaboration system for augmented maps by placing four AR markers near the printed map [17]. The user can draw annotation on a paper map by using digital pen and share their modifications with the other users. However, AR toolkit is not robust against occlusions. The virtual contents are also limited to predefined data. On top of that, the marker obstructs the appearance of the map.

Many approaches tried to develop a marker that is visually related to the system. Some attempts used a map as the marker and extract its features for matching such as utilizing SIFT feature detector [15]. Recently, a fast keypoint detection using random ferns had also been developed for map tracking [21]. Another approach used mutual information between two map images for tracking [9].

Reitmayr et al. developed augmented maps used natural features tracking in table-top system equipped with a projector [23]. Their system could project the additional information on top of the map. Moreover, the user can select information using PDA device as a pointer. Similarly, Rohs et al. used patches in a paper map as the visual descriptor for detection and tracking [24]. Furthermore, they used mobile devices for displaying additional information.

In contrast to the texture-based map tracking above, keypoints based tracking has been also explored. Nakai et al. used random keypoints as features for camera pose estimation [20]. Their work fails on matching surface in extreme camera tilt because they do not use the previous successful tracking information. Their work had been improved by Uchiyama and Saito in so called the tracking by descriptor update [25]. The method updates the descriptor database and successfully tracks a paper in extreme camera tilt. By using the same method, they had developed augmented maps with intersections in a map as keypoints [27]. They colored the keypoints and extracted them using color detection. They introduced the random dots marker that successfully achieves robust and accurate tracking using thousand markers [26]. The random dots marker is also previously applied in the development of the foldable augmented maps [16].

In this research, we utilize the random dots for detecting and tracking the maps. Specifically, we can generate many dots from geographical data such as distribution of buildings and landmarks in a map. We assume that buildings and landmarks are distributed randomly over the map. As a result of using the random dots, we can track the map while preserve the appearance of the maps because the dots are relevant information in the map.

Besides tracking, researchers have drawn attentions to the collaboration on augmented reality. Generally, augmented reality application contains predefined 3D models which are not reusable for another application. Instead of creating 3D models from scratch, some AR applications retrieve them from the Internet. For instance, AR sights system allows users to download available markers from its website and 3D models from Google 3D Warehouse [1].

Live videos augmentation on aerial map was explored by Kim et al. [14]. They also applied the real lighting condition estimated directly from the maps image. Similarly, Bing from Microsoft integrates maps, panorama pictures, and live videos submitted by users in one spatial augmentation [3]. Both works use collaboration among users. However, they only augment on the digital map instead of paper maps.

Similar to our approach, Morrison et al. used the natural features to track a printed Google map and visualized additional information on top of it [19]. However, they did not augment the 3D city models onto the printed map. Similarly, Paelke et al. integrated a paper map and additional information on mobile devices [22]. Gruber et al. provided a dataset for tracking using city models from Google 3D warehouse [10]. However, their work only covered virtual contents preparation and omitted map detection and tracking.

All approaches above used conventional way where the user has to prepare the map and the virtual data separately and connect them manually. To solve these issues, we propose an integrated data preparation procedure for retrieving shared 3D models from Google 3D warehouse together with the map image from Google Maps. Since they are in the same geographical coordinate, both are automatically connected. Therefore, we can augment the 3D models directly onto the paper map. Furthermore, we can use the map for tracking as replacement of fiducial marker. Since we can query any location in Google Maps, as a result, we can build the world augmented maps using our approach.

3 Proposed System

Our proposed system focuses on the data access to maps and 3D models database for realizing the augmented maps. We also explore the suitable tracking method for the maps. In our system, we use Google Maps and 3D Warehouse as our resources. The flow of our proposed system is illustrated in Fig. 2.

3.1 Map Data

We retrieve maps from the Google map. Three types of map are available: default, terrain, and satellite maps as illustrated in Fig. 3.

3.1.1 Map production

We create a map by combining 3D model locations as a tracking layer on top of a background map as illustrated in Fig. 4. We define the tracking layer as colored dots. This layer is then extracted back in the initialization step using color detection.

We create a tool for retrieving the produced maps (tracking layer + background map) from Google Maps and 3D models from 3D Warehouse as illustrated in Fig. 5. Our tool receives a city name as the input and displays locations of 3D models on top of the background map the city as colored dots. The right panel beside the map enlists the name of the model appears on the map. The user can download the local database of the city map with the associated 3D city model. The output of our tool are a trackable map, a set of 3D building models inside the map area and a text file contains the position of 3D building in geographic coordinate. The text file is used for detecting the paper map in the initialization step 3.2.1.

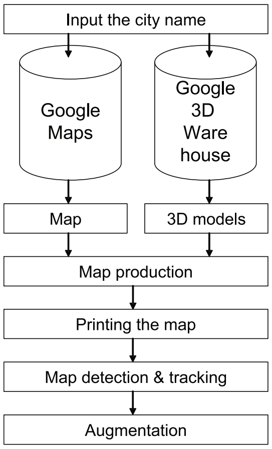


Figure 2: Flow of the proposed system. First, the user inputs a city name. The system requests to servers for a map and 3D models. The trackable map is then made in the map production. The user then prints the map. At online phase, the paper map is detected and tracked. The 3D models are then augmented.

3.1.2 3D Data

Google 3D Warehouse allows users to create and share their 3D models. Since all 3D models in Google 3D Warehouse are made in geographic coordinate, they can be augmented directly onto map that also uses geographic coordinate. For our system, we only use 3D models of buildings and landmarks in a city. Currently, the user downloads the maps and 3D models using our tools beforehand. However, in the future, the 3D models can be downloaded simultaneously when the application runs.

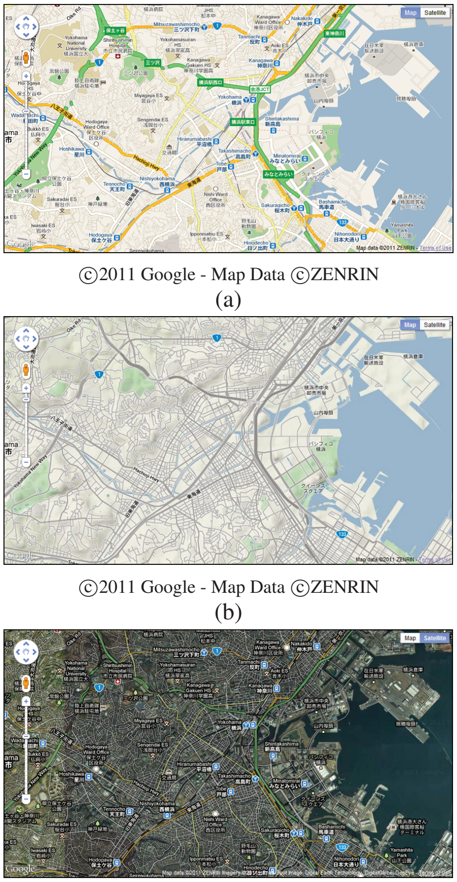
3.2 Map Tracking

For augmenting 3d models on top of a map, a camera pose estimation is necessary because the view should follow the orientation and the movement of the camera. One way to estimate camera pose is placing a planar marker on the map and estimate it using homography. Thus, whenever the markers is detected, the camera pose is estimated and the 3D models can be rendered correctly.

However, this fiducial marker approach is not robust against occlusion. It also obstructs the appearance of the map. Therefore, we utilize the random dots marker approach because it is robust against occlusion and the random dots marker can be generated from the geographical data. First, we prepare keypoints database that includes the location of the 3D models. By using the matching method in the random dots marker method, the geographic coordinate of the map is acquired. Then, we start to track the map using the random dots marker or another method such as SIFT, SURF or random ferns.

3.2.1 Initialization

The random dots are generated from the 3D models location from the Google 3D warehouse. We use these dots as the keypoints for matching. These keypoints are then stored to a file that can be loaded when the system starts as illustrated on Fig. 6. The initialization is the key of the coordinate transformation from the geographic coordinate to image coordinate. Since the dot markers can be retrieved in both coordinates using Google API, then the transformation from image to geographic coordinate can be estimated. By knowing the geographic coordinate, we can load the 3D models that exists near the position.



©2011 Google - Map Data ©ZENRIN

(a)

©2011 Google - Map Data ©ZENRIN

(b)

©2011 Google - Imagery ©2011 Cnes/Spot Image, Digital Earth Technology, DigitalGlobe, GeoEye, Map data ©2011 ZENRIN

(c)

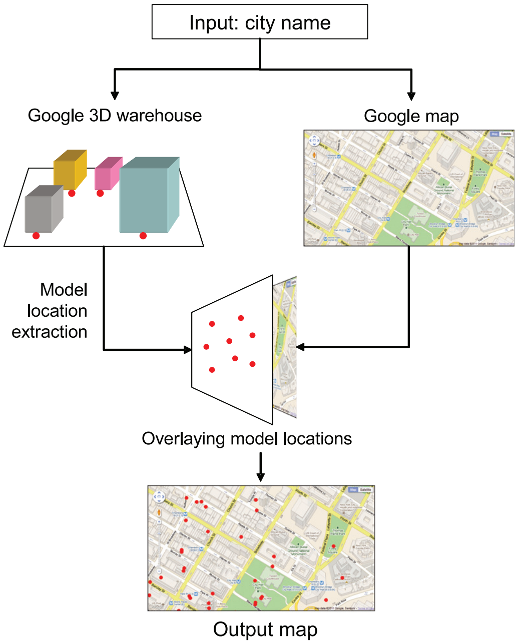
Figure 3: Three types of Yokohama map. (a) Default map. It consists some labels. (b) Terrain map. It consists dense edges and lines. (c) Satellite map. The real captured image from the satellite.

Keypoints are matched in the initialization step as illustrated in Fig.7. In offline phase, we create a descriptor database from the text file that contains the location of buildings (dots) in the geographic coordinate. The descriptor of a point are computed by estimating its relationship to the neighboring points.

In online phase where a camera captures the printed map, we extract dots in tracking layer and output a binary image. We then calculate the descriptor for each dots followed by matching the calculated descriptors with the descriptors in the database. At this stage, many matches are established and the map is detected.

3.2.2 Tracking

After the matches are established in the initialization, we start to track the map on succeeding frames. For tracking using random dots markers, we only updates the descriptor database using descriptors calculated in current frame. For tracking, we also apply another tracking methods such as SIFT, SURF and random ferns. In contrast to random dots marker that uses keypoint features, SIFT, SURF and random ferns use the texture of map image for matching. Theoretically, all four methods are applicable for augmented maps. However, each method requires different initialization. For instance, creating the index of the keypoints database is necessary for the random dots marker. Whereas random ferns method requires a learning process. The preparation time varies to each method. We compare the results of each method in terms of the number of suc-



©2011 Google - Map Data ©2011 Google, Sanborn

Figure 4: Map production flow. First, a city name is queried. The tool extracts the map and 3D models of the city. The local database for initialization are then built. The 3D model positions are overlaid as the colored dots.

cessful tracking and the computational cost in Sec. 5.

4 Interaction for the Augmented Fly-Through Application

We develop an augmented fly-through that allows user to browse and view the 3D city models through camera or HMD on a paper map. The user also can select the information on the map, by moving the camera as a pointer (see Fig. 8). We prepare the building name as the virtual information that will appear when the center of the camera approaches the models.

5 Results and Evaluation

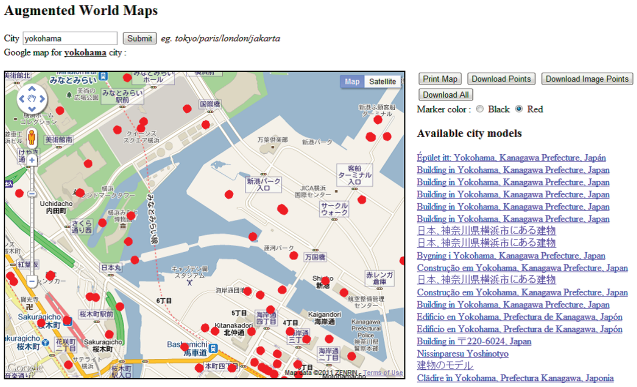
The outputs of our system are the augmentation of any locations in the world as illustrated in the Fig. 9. The virtual contents in our system fully depends on the models availability in Google 3D warehouse.

For tracking, we apply four detection and tracking methods: random dots markers, SIFT, SURF, and random ferns. We count the successful tracking based on three types of map: default, terrain and satellite map. We also estimate the computational cost for convincing that our system runs in real time. For our experiments, we use a web camera with resolution 640×480 and we print the map in A4 size paper.

We implemented our system using OpenCV library [6]. The city models are loaded using Open Asset Import Library (Assimp)[2]. We calibrate the camera using the Calibration tool [4] that is based on the implementation of Zhang calibration method [28]. For experiments, we use a laptop computer with specifications: Intel I7 Quad Core 2.80GHz and 4GB memory.

5.1 Tracking Robustness

In this section we show the percentage of successful tracking on image sequences contains the image map. For this experiments, we


Augmented World Maps

City yokohama Submit *eg. tokyo/paris/london/jakarta*
Google map for yokohama city :

Marker color : ○ Black ● Red

Available city models

Épület itt: Yokohama, Kanagawa Prefecture, Japán
Building in Yokohama, Kanagawa Prefecture, Japan
Building in Yokohama, Kanagawa Prefecture, Japan
Building in Yokohama, Kanagawa Prefecture, Japan
Building in Yokohama, Kanagawa Prefecture, Japan
Building in Yokohama, Kanagawa Prefecture, Japan
日本, 神奈川県横浜市にある建物
日本, 神奈川県横浜市にある建物
Bygning i Yokohama, Kanagawa Prefecture, Japan
Construção em Yokohama, Kanagawa Prefecture, Japan
日本, 神奈川県横浜市にある建物
Construção em Yokohama, Kanagawa Prefecture, Japan
Building in Yokohama, Kanagawa Prefecture, Japan
Edificio en Yokohama, Prefectura de Kanagawa, Japón
Edificio en Yokohama, Prefectura de Kanagawa, Japón
Building in 〒220-6024, Japan
Nissinparesu Yoshinotyo
建物のモデル
Clădire în Yokohama, Prefectura Kanagawa, Japonia

©2011 Google - Map Data ©2011 ZENRIN

Figure 5: A tool for extracting city map (Yokohama) from Google Maps and 3D warehouse. The red dots on the map represent the location of 3D building models. The list on the right panel shows the available 3D building models.

use Kyoto map that consists 84 mesh models. Accordingly, we have 84 dots on the map. We prepare three image maps: default, terrain and satellite maps. We capture those three maps using web camera and record them as image sequences. We then applied each tracking method on the image sequences. Successful tracking occurs if the system can detect the map on the image sequences. We then reproject the border of the map using the homography to the map image. We count the frame of which the projected border is near to the actual border of the map in the paper divided by the number of frames as illustrated in the Fig. 10.

According to the results, texture-based tracking using SIFT, SURF and random ferns are robust on default and terrain maps. We note that default and terrain map have strong edges and distinctive colors that help the successful tracking. On the other hand, the robustness of tracking drops on the satellite maps. In the satellite maps, the texture is relatively uniform that makes the matching becomes difficult.

As we expected, the random dots marker method could track the paper maps regardless the type thanks to the tracking layer. Surprisingly, the tracking robustness even increased on the satellite maps . We realize that the dots become distinctive enough on the satellite map to make them easy to extract.

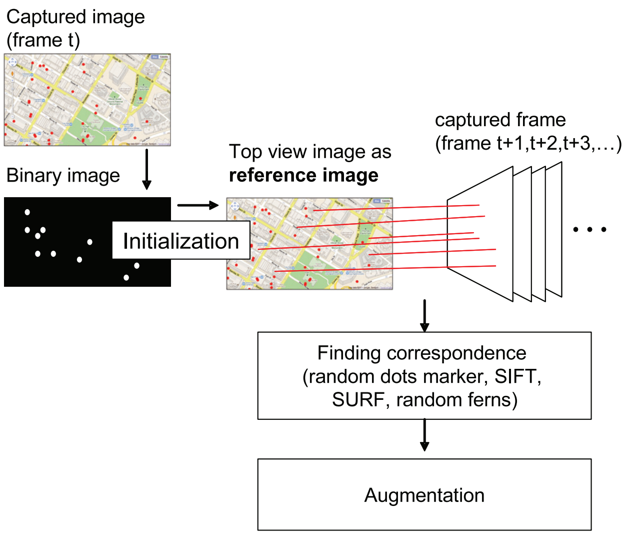
5.2 Computational cost

We evaluate the computational cost of our application on the matching and rendering process. First, we provide a map with some models for augmentation. We compute the average time for matching the map with the reference map and rendering the models as listed in Table. 1.

Table 1: Computational cost based on the type of the map and tracking method.

Method	default map (ms)	terrain map (ms)	satellite map (ms)
Random dots marker	45.41	52.9	51.49
SIFT	853.39	844.91	926.02
SURF	462.61	440.06	827.46
Random ferns	174.26	168.44	202.14

We can see from the results that random dots method works faster than the other method because it depends only on the color



©2011 Google - Map Data ©2011 Google, Sanborn

Figure 6: From initialization to tracking. First, a frame is captured. Each frame is binarized and the keypoints are extracted using color detection. The descriptors are then computed and matched with the keypoints database. When the frame is matched, the homography is computed. The frame is warped using inverse homography to get the top view of the map image as reference for tracking.

detection. In addition, it uses a hashing technique for fast descriptor lookup. On the other hand, the matching method that utilizes the texture information such as SIFT, SURF and random ferns requires longer time. SURF has better performance than SIFT thanks to the integral image approach. Random ferns works best among the three methods. However, random ferns requires around 10 minutes learning or building database beforehand. This 10-minutes-long learning is not suitable for our purpose because preparing learning data every time the user download the map will make our application unpractical.

6 Discussion

In our experiments, tracking using SIFT, SURF and random ferns can work robustly for default and terrain maps. We can choose those method if we want to use default map and terrain maps. On the other hand, if we want to use the satellite maps, random dots marker can be the alternative for tracking.

Moreover, random dots marker contributes less time than the other method thanks to the simple extraction method and hashing. The computational cost is significant for deciding the suitable tracking method for augmented maps. Comparison of another feature descriptor for tracking paper map such as BRIEF [8], or GLOH [18] and its variant are the next step of this research. Furthermore, it is interesting to explore on combining the random dots marker with the other texture based method for realizing the best tracking method for augmented maps.

Technically, we add a tracking layer on a background map for initialization. This solution is practical to implement. Currently, we create the tracking layer by overlaying colored dots over the background map. In order to create more realistic map, instead of colored dots, we can print colored icons.

On the other hand, instead of adding tracking layer on the background map, we are interested on different approach by extracting specific features on the original map. Since there are terrain and satellite map, feature extraction will be different for each type and

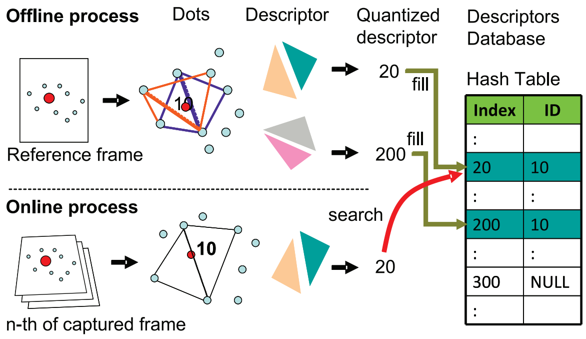
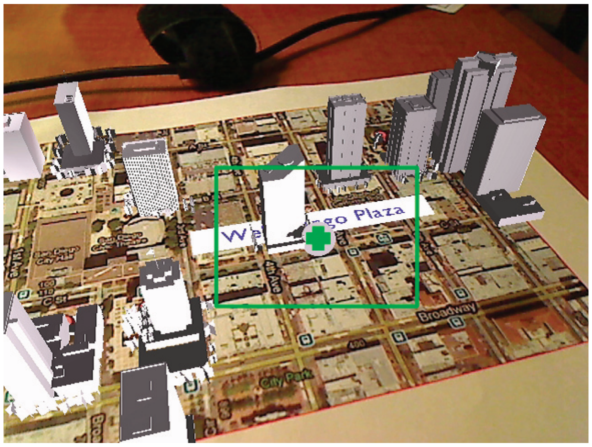


Figure 7: Keypoints matching. There are two main process on the matching. Offline process estimates descriptors of all dots/keypoints and store them to a descriptor database as the index of dots id. The online process extract dots from captured image and compute the descriptor of the dot followed by searching the dots id based on the descriptor.



©2011 Google - Imagery ©2011 Google, Map data ©Google

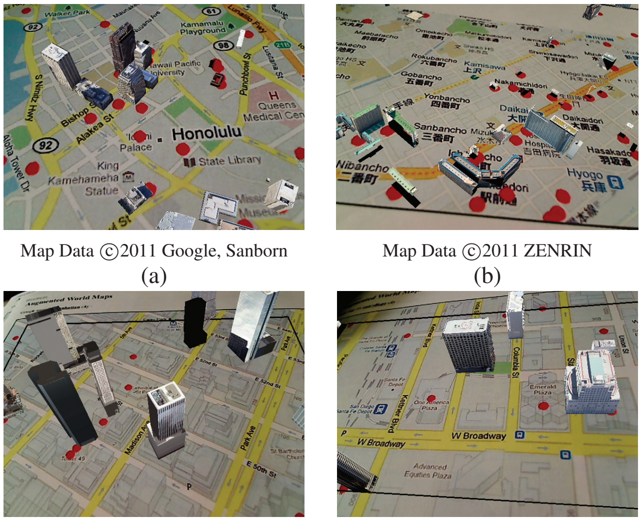
Figure 8: Data selection using center of the camera image. The user moves the camera to select and display the information of the San Diego map. The name appears when the center of the camera image approaches a building model.

we lose the generality for each type of map. However, defining features for particular type of map is interesting issue to explore.

We are also interested on bringing this research more on the collaboration aspects. Through our application, we have shown that the geographic coordinate can be utilized as the shared space for augmented reality. However, our current implementation only covers the viewing workspace of the collaboration. To add the individuality features for collaborative AR environment [12], it is necessary to let user to create and modify the shared 3D models on the augmented maps. Thus, each user can view coherent 3D models on their site.

7 Conclusion

We have presented our augmented fly-through using shared geographical data. The users can use and share virtual contents from the maps and 3D model database server. Therefore, they can view any location in the world through augmented reality. We have also presented our study on the characteristics of map and its usage for developing augmented maps. Finally, we proved that random dots



Map Data ©2011 Google, Sanborn
(a)

Map Data ©2011 ZENRIN
(b)

Map Data ©2011 Google
(c)

Map Data ©2011 Google, Sanborn
(d)

©2011 Google

Figure 9: Augmentation results. 3D building models are augmented on top of printed maps. (a) Honolulu (b) Kobe (c) Park Avenue (d) San Diego.

marker is suitable for our system in terms of the minimum computational cost.

Providing an instant way for retrieving the maps and virtual data for augmented maps remains as our main challenges. In the future, on-line connection to the database server and the cloud architecture are promising outlooks to access the maps and virtual data efficiently. As a result, we can achieve the robust and rich augmented maps application. We will improve our user interaction by implementing robust finger detection. We will also work on occlusion handling and realistic rendering for augmented maps.

Acknowledgement

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for high-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan and Grant-in-Aid for JSPS Fellows. We also thank Hideaki Uchiyama for providing the random dots marker source code.

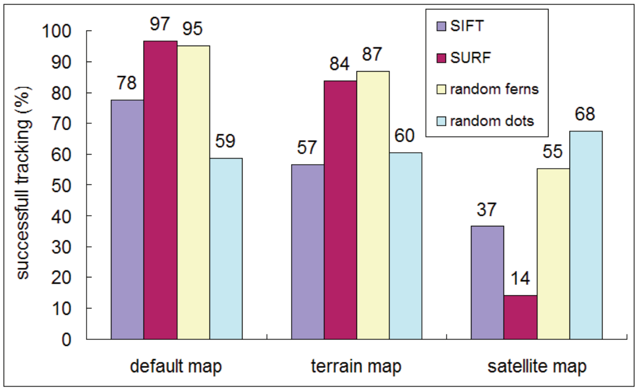


Figure 10: Successful tracking rate. Random dots marker works on any type of map.

References

[1] AR Sights. http://www.arsights.com/.

[2] Assimp. http://assimp.sourceforge.net/.

[3] Bing Maps. http://www.bing.com/maps/.

[4] Camera Calibration Tools. http://www.doc.ic.ac.uk/ dvs/calib/main.html.

[5] Google Map and 3D Warehouse. http://www.google.com/.

[6] OpenCV. http://sourceforge.net/projects/opencvlibrary/.

[7] J. Bobrich and S. Otto. Augmented maps. In *ISPRS*, 2002.

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proc. ECCV*, September 2010.

[9] A. Dame and E. Marchand. Accurate real-time tracking using mutual information. In *Proc. ISMAR*, pages 47–56, 2010.

[10] L. Gruber, S. Gauglitz, J. Ventura, S. Zollmann, M. Huber, M. Schlegel, G. Klinker, D. Schmalstieg, and T. Höllerer. The City of Sights: Design, construction, and measurement of an Augmented Reality stage set. In *Proc. ISMAR*, pages 157–163, 2010.

[11] N. R. Hedley, M. Billinghurst, L. Postner, R. May, and H. Kato. Explorations in the use of augmented reality for geographic visualization. *Presence*, 11:119–133, 2002.

[12] A. Ismail and M. Sunar. Survey on collaborative ar for multi-user in urban studies and planning. In *Learning by Playing. Game-based Education System Design and Development*, volume 5670 of *Lecture Notes in Computer Science*, pages 444–455. 2009.

[13] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proc. IWAR*, pages 85–94, 1999.

[14] K. Kim, S. Oh, J. Lee, and I. Essa. Augmenting aerial earth maps with dynamic information. In *Proc. ISMAR*, 2009.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[16] S. Martedi, H. Uchiyama, G. Enriquez, H. Saito, T. Miyashita, and T. Hara. Foldable Augmented Maps. In *Proc. ISMAR*, pages 65–72, 2010.

[17] D. McGee, X. Huang, P. Barthelmess, and P. Cohen. Poster: The neteyes collaborative, augmented reality, digital paper system. In *Proc. 3DUI*, pages 145 –146, March 2008.

[18] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005.

[19] A. Morrison, A. Oulasvirta, P. Peltonen, S. Lemmela, G. Jacucci, G. Reitmayr, J. Näsänen, and A. Juustila. Like bees around the hive: a comparative study of a mobile augmented reality map. In *Proc. CHI*, pages 1889–1898, 2009.

[20] T. Nakai, K. Kise, and M. Iwamura. Camera based document image retrieval with more time and memory efficient LLAH. In *Proc. CBDAR*, pages 21–28, 2007.

[21] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *TPAMI*, 32:448–461, 2010.

[22] V. Paelke and M. Sester. Augmented paper maps: Exploring the design space of a mixed reality system. *ISPRS*, 65:256–265, 2010.

[23] G. Reitmayr, E. Eade, and T. Drummond. Localisation and interaction for augmented maps. In *Proc. ISMAR*, pages 120–129, 2005.

[24] M. Rohs, J. Schoning, A. Kruger, and B. Hecht. Towards real-time markerless tracking of magic lenses on paper maps. In *adjunct Proc. Pervasive*, pages 69–72, 2007.

[25] H. Uchiyama and H. Saito. Augmenting text document by on-line learning of local arrangement of keypoints. In *Proc. ISMAR*, pages 95–98, 2009.

[26] H. Uchiyama and H. Saito. Random dot markers. In *Proc. IEEE VR*, pages 35 –38, march 2011.

[27] H. Uchiyama, H. Saito, M. Servieres, and G. Moreau. AR GIS on a physical map based on map image tetrieval using LLAH tracking. In *Proc. MVA*, pages 382–385, 2009.

[28] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proc. ICCV*, pages 666–673, 1999.