AR Representation System for 3D GIS based on Camera Pose Estimation using Distribution of Intersections

Hideaki Uchiyama, Hideo Saito
Keio University
3-14-1 Hiyoshi, Kohoku-ku 223-8522, Japan
{uchiyama,saito}@ozawa.ics.keio.ac.jp

Vivien Nivesse*, Myriam Servières†, Guillaume Moreau‡
Ecole Centrale de Nantes - CERMA IRSTV
rue de la Noë, BP 92101, 44321 Nantes Cedex 3, France
vivien.nivesse@eleves.ec-nantes.fr, {myriam.servieres,guillaume.moreau}@ec-nantes.fr

Abstract

This paper presents a framework for overlaying 3D GIS data information onto a 2D physical urban map. Such data may be 3D buildings, soil composition or animations, displayed in real-time with a moving camera. We propose a map recognition framework by topological information in order to recognize the area of the physical map from a whole map and display its 3D data. The retrieval of the geographical area described by the physical map is based on a hashing scheme with local combinations of intersection points, retrieved with a voting procedure from a previously computed index. Specific features are then tracked to allow the overlay in real-time.

The results show that augmentation of physical maps based on topological features only is possible, allowing the use of any physical map without the need for specific markers, and providing the user with intuitive navigation.

1. Introduction

Geographical Information Systems (GIS) have become essential tools for local authorities for studying, handling and planing urban development. GIS can be seen as tools that superimpose layers (representing homogeneous information) that are fused together to generate maps. GIS data can be updated any time and are thus more up-to-date than traditional paper maps. They can moreover be adapted in real time to meet the user's need.

Previous works [6, 15] have shown the advantages of using Augmented Reality techniques to display digital information on standard paper maps, because such maps are easier to manipulate. Moreover, GIS need a shift towards 3D to be compatible with sustainable development concerns: To manage increasing complexity of sustainable development requirements, 3D+t queries have to be handled to compute new indicators that are now being defined. A thermal comfort indicator could be for instance 'walls that have more than 8 hours sunlight in winter and less than 2 hours in summer'. Visualizing the results of such a query requires 3D because sunlight exposure is dependent on building height and neighboring buildings. 3D virtual environments are not easy to manipulate for local authorities, that is why we assume that the use of AR maps will facilitate the display of such results by letting the user manipulate both a paper map and the viewpoint in a natural way.

In this paper, we propose a framework of map recognition technique to establish a correspondence between the image of a real map captured with a camera and a GIS. Specific features are extracted from the input image, then matched with the GIS data, as in the problem of *Document image retrieval*. We are then able to compute the camera position with respect to the map, and display more information from the GIS. The evaluation of our system will include AR representation and its computation cost. The comparison between related works won't be included because we still have some works to a real map.

The rest of the paper is organized as follows: we will first briefly present related works that can be used to match images representing the same objects, i.e. compute the geometric transformation that links the two images. We will then provide an overview of our system (section 3.1) that

*Ecole Centrale de Nantes
†Ecole Centrale de Nantes - CERMA IRSTV
‡Ecole Centrale de Nantes - CERMA IRSTV

ICAT 2008
Dec. 1-3, Yokohama, Japan
ISSN: 1345-1278

requires an initialization phase (section 4) that allows a first camera pose estimation (section 5) and a tracking phase (section 6). Finally, experimental results will be presented and discussed in section 7.2.

2. Related works

The problem of finding a match for a query object using feature points has been addressed in various ways. The feature points can be described using rich descriptors such as SIFT [10], PCA-SIFT [8] or SURF [2], that typically use image patches that are robust in terms of change of illumination, scale and rotation and describe them with high-dimensions vectors. The search methods have then to deal with the problems of nearest neighbor search in high dimensions with efficient algorithms [1], locality-sensitive-hashing [4] or a vocabulary tree [13].

Rich descriptors are well suited to the retrieval of images near-identical to the ones in the database, with few repetitive texture patterns. By contrast, 2D maps can be presented in different ways, according to the manufacturer, and the retrieval method needs then to focus on the geometry of the urban environment they describe. For this reason, the feature points need to be specific to urban environments, and the location of the roads' intersections are used in this paper.

It is not possible to make a database query using only the location of a single feature point, so the essential information in retrieval is the arrangement of the features points. Such an arrangement, in our case, must be invariant to the orientation of the camera relative to the map.

Geometric hashing (GH) is such a general model-based object recognition method [9, 17] widely used in computer vision as well as in other domains such as bio-informatics [14]. The introduction of a geometric invariant yields a computational cost quite important, in $O(N^5)$, that is unsuitable for an augmented reality application. A probabilistic reduction of the number of feature points results in an accuracy degradation and has led to the introduction of "locally likely arrangement hashing" (LLAH), which outperforms GH in both processing time and required amount of memory [11, 7]. In this scheme, neighboring points are considered for the calculation of an affine invariant used as a key in a hashing table. A voting technique is employed for retrieval, insuring efficiency and robustness against erasure of feature points.

We use a combination of this method and a more traditional tracking technique to first recognize the area in the camera filed of view, then overlay 3D buildings in real-time.

3. Overview

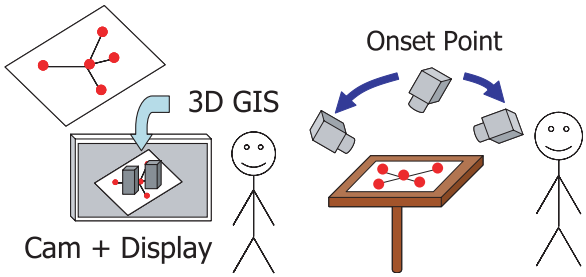
3.1. System

The user is handling a hand-held device equipped with a camera coupled with a computer, for example a cellular phone or a see-through HMD. In our experimental setup, we use a digital camera and a laptop (see in Figure 1(a)).

The physical map can be displayed on a desktop, on a wall or any flat surface. There are no requirements on the manufacturer or the map edition, since we focus on the topological relations between features and not on extrinsic properties.

For a proper initialization, the camera needs to be in a position more or less parallel to the map, so that perspective distortion is not too important . Once the tracking stage begins, the user can move more freely, he can even position the camera at important angles (see in Figure 1(b)). Even though the tracking process is robust, it can fail after a fast movement or acquiring a small area of the physical map. After such an event, the system will try to re-initialize and the user has to resume a proper camera position.

The 3D buildings and the GIS data are displayed in real-time on the screen of the device, providing the user with additional information.



(a) Setting (b) Usage

Figure 1. System Overview

3.2. Algorithm

The framework proposed here is based on LLAH as a map recognition technique by using topological information. A top view image of the 2D map is acquired and processed, and individual feature points are recognized with the main LLAH framework as a retrieval process in the initialization stage (see in Figure 2(a)). If enough points are correctly recognized, the homography between the image plane and the camera can be computed (see in Figure 2(b)) and the tracking stage begins (see in Figure 2(c)). LLAH is not used during the tracking stage because of the computation needs of that method, and its poor accuracy under strong perspective distortion.

Once the homography is computed, a query is made to the database to try to identify all points present in the image,

allowing to select points with few neighbors, that will be easy to track. Once good points to track are localized, the next image can be checked from them. If enough of them are found, they are then used with a RANSAC [5] process to compute the homography and the 3D data can be overlayed. The tracking may then repeat itself on the next image.

If the tracking fails, because the camera positions are too different between two successive images, or not enough points are present on the image, the initialization process is repeated.

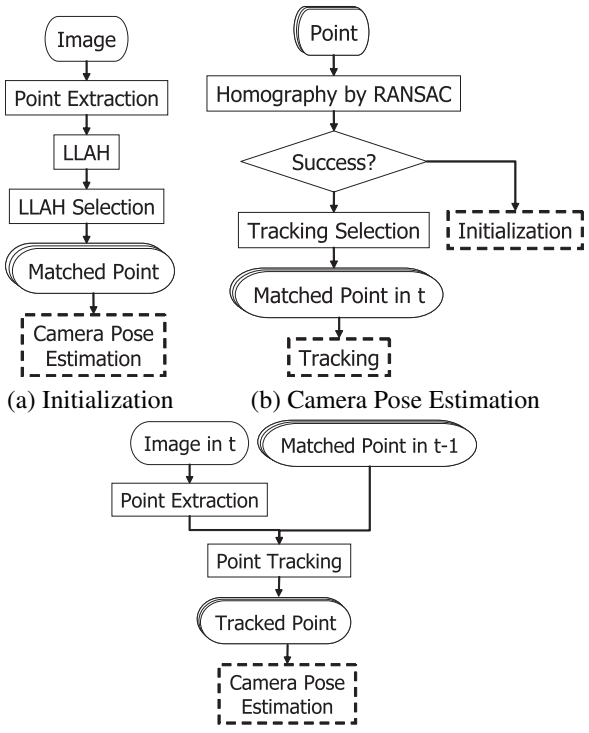


Figure 2. Algorithm Overview

3.3. LLAH

3.3.1 Overview

LLAH [11, 7] describes a method for indexing and retrieving specific feature points in documents based on only local arrangements, allowing for partial occlusion and a degree of perspective distortion.

In the off-line stage, LLAH considers specific entities of an image database called feature points each at a time, extracts different sets of points from its immediate neighbors to calculate perspective invariants, then for every set creates the corresponding entry for that point in a hash table. In the retrieval stage, the same process is applied to the input image. Since the keys to the hash table are computed from the invariants, it is possible to access the ID of the point from that knowledge alone. For each point, there are as many accesses to the hash table as possible calculation of invariants for every set of neighboring point, so a voting process casts a vote for every retrieved ID and minimizes the impact of registration errors.

3.3.2 Indexing

The off-line stage creates a hash table from the input data. The first step is the extraction of every feature point, a feature point being a specific entity of the input image, for example word centroids in text documents or SURF features in an image. For every feature point, we consider some subsets of its closest neighbors. The reason subsets are interesting is that the distance between points may be impacted by perspective projection, but some subsets will remain the same for at least a given degree of perspective. A good affine invariant is the ratio of the area of two triangles drawn from four points (see in Figure 3). We use subsets of size 5, so there are five ways to compute the affine invariant. These five invariants are quantized to reduce the registration errors, then used as parameters in a hashing function in order to compute a hash key, that is used to store the ID of the considered feature point, as well as the order and value of the invariants. The process is repeated for every subset of five points from the seven closest neighbors to the feature point. These values, five and seven, have been estimated in [11] as the best trade-off between robustness to perspective projection and computation requirements. Once the process has been completed, the hash table contains numerous entries of points' IDs and the corresponding lists of invariants.

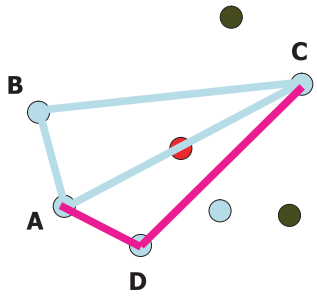


Figure 3. Affine Invariant

3.3.3 Retrieval

The retrieval process aims at correctly identifying individual feature points from an input image. For this, the exact same process as in the indexing stage is carried out for each extracted feature point, up until the point when the hash key is computed. When the hash key has been computed from invariants, the corresponding bin in the hash table is retrieved and the exact order of the invariants that have been

stored there is checked. This removes the collision issues and ensures that the match is correct. If the orders are identical, a vote is cast for the stored ID. Once every subset of five points from the seven closest neighbors has been used to cast a vote, the ID with the most votes is mapped to the extracted feature point. In this way, it does not matter if there are registration errors as long we can assume that they follow a gaussian distribution: after the hashing process, the votes cast for false ID will be evenly distributed and only the correct ID will receive a significantly higher number of votes. This process performs very well even for very large databases (10000 documents in (reference)), but the recognition accuracy fares poorly under strong perspective, as the affine invariance assumption is no longer valid.

3.4. Database creation and segmentation

Real GIS data of a large French city is used. GDMS [3] is used to process the data in two ways:

- with a simple query, all intersections are extracted from the road network to build the features points that are used in the method.
- following Neubauer and Zipf's idea [12], we have built an XML style file that describes how the GIS database will be rendered in the virtual environment, i.e. whether a polygon layer should be rendered with flat surfaces or extruded polygons, and additional informations such as the color to use. We have thus built a VRML builder above GDMS that transforms GIS data according to the XML file and generates a VRML file.

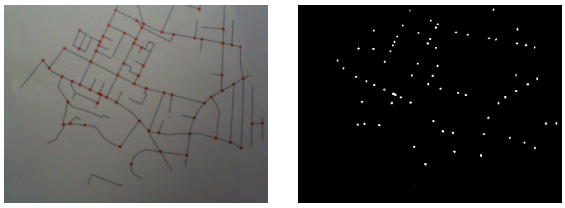
The area described by the data can in theory be very large, and must be sub-divided in sub-areas that correspond to the size of the physical maps used as queries. These sub-areas are defined by a specific ID that is stored alongside the ID of the feature points in the database.

The registration step of the LLAH algorithm can then be applied, resulting in the creation of the hash table.

4. Initialization

4.1. Point Extraction

The feature points extraction is the process applied to the input image that extracts the road intersections of the map. Ultimately, the extraction will be done automatically, but as a proof of principle, the feature points are tagged manually on the physical map with color dots (see in Figure 4(a)). The acquired image is converted from RGB to HSL space, then thresholded to retain only the specific color used (see in Figure 4(b)).



(a) Input (b) Color Extraction

Figure 4. Point Extraction

4.1.1 ID retrieval by LLAH

The algorithm of locally likely arrangement hashing is applied on the points extracted from the input image. As the process is sensitive to perspective distortion, the input image needs to be close to a top-view image for optimal accuracy. The stronger the skew of the camera relative to the physical map, the less points are correctly recognized.

The area described by the map can be recognized by a voting on the origin area of the recognized points (see in Figure 5).

As described in 3.3, one point is recognized from several neighbors. Since it is necessary to have more than four points for calculating a homographpy, many points should be captured in the image (in Figure 5, more than 50 points).

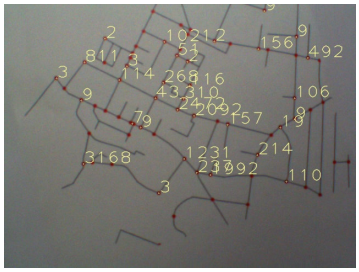


Figure 5. ID retrieval by LLAH

4.1.2 LLAH Selection

During the LLAH process, the extracted points are set to an ID that corresponds to the ID collecting the most votes from accesses to the hash table. In this paper, we present a rough method for estimating the confidence with which these pairings are set. By accessing the voting table like Table 1, we can sort the points according to the number of votes that were cast in the best bins. A point that is attributed an ID with a high number of votes is more likely to be correctly recognized than a point with a low number of votes. For this reason, we only keep a given number of pairs N, that are the N pairs with the most votes on the ID they were attributed (in our case, N=10). Points in Figure 6 are selected from Figure 5.

This culling of the number of recognized points prevents

false match from entering the RANSAC algorithm during the camera pose retrieval process.

Table 1. Voting Result

1231	341	43	499	86
40	20	6	6	3

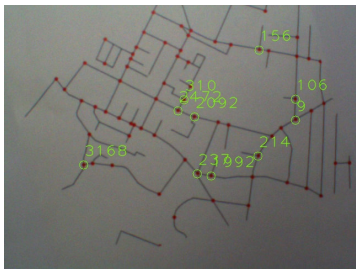


Figure 6. LLAH selection

5. Camera Pose Estimation

5.1. Homography Calculation by RANSAC

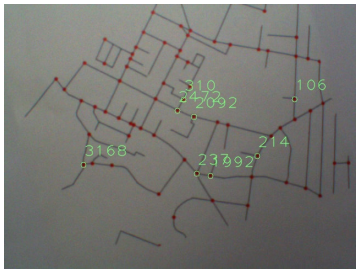
As mentioned in 4.1.2, N points with a high number of votes are selected. Whilst a good indicator of the confidence with which a point is matched, a high number of votes may nevertheless occur with a false match, because there exist similar distributions of intersections in a map. For this reason, the calculation of the homography is based on the RANSAC algorithm, for its ability to cope with false matches without compromising the accuracy of the homography.

The process begins by a random selection of 4 points out of the N best matches, which is the minimum number of points required for the calculation of the homography. If $N = 4$, the RANSAC process can't work and is terminated, a new initialization taking place afterwards.

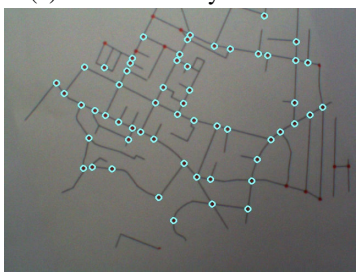
A first homography between the camera plane and the map is computed from the randomly sampled 4 points. To evaluate the homography accuracy, all N points extracted from the map are re-projected onto the image plane to compute reprojection errors. The number of points for which the reprojection error is less than a threshold (in our case, 3 pixels) are counted and stored. The random sampling and its evaluation are repeated several times (in our case, 50 times), then the four points for which the computed homography had the smallest reprojection error are selected. Figure 7(a) shows an example of selected points by RANSAC from Figure 6.

The number of points for calculating the homography is less than N as shown in Figure 7(a). To get more matched points for calculating the accurate homography, the data points from the GIS are re-projected onto the image plane. If the re-projection error is less than a threshold (in our case, 3 pixels), the point will be a matched point. Compared with Figure 7(a), there are more points in Figure 7(b) and the homography is re-calculated by using these points.

Since a homography includes components of camera position and orientation, a projection matrix is calculated for the overlaying of GIS data [16].



(a) ID selection by RANSAC



(b) Re-projected Points

Figure 7. Homography Calculation by RANSAC

5.2. Tracking Selection

After the homography calculation has been successfully done, good points to track are selected for the tracking phase. Since the color of all intersections is identical, template matching methods are not appropriate for matching points from different views. For this matching phase, we use the assumption that the change in viewpoint between successive frames is small, an assumption commonly made in AR applications.

Fig. 8 shows the extracted feature points in the image at $t-1$(circle) and in the image at t (square). The points extracted in the image at t are matched to the nearest points extracted in the image at $t-1$. For example, circle c is matched to square f and circle b is matched to square e. However, the matching between circle b and square e is false, as it should be matched to square d. This shows that if the distance between points is adopted as the criteria for matching, false matchings happen when several neighbor points exist.

In order to prevent false matchings, we select points that are not too close to the image borders nor to their neighbors. If a point is close to an image border, there is a possibility that it will be out of sight in the next frame. If a point is close to its neighbors, a wrong matching may occur as

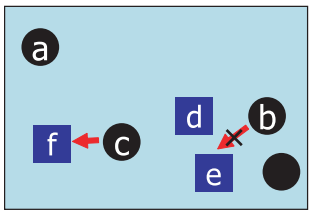


Figure 8. Matching Points

mentioned above. By defining thus good points to track, the tracking process is more robust and accurate.

For each point in Figure 7(b), the distances between the nearest neighbor and the map borders are calculated and sorted. Afterwards, the N points that have the longest distances are selected as good points to track (Figure 9).

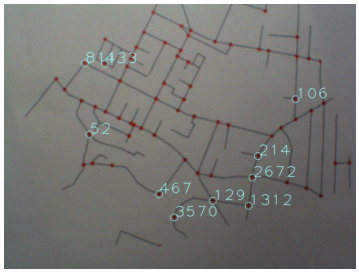


Figure 9. Tracking Selection

6. Tracking

In the tracking phase, extraction of features from an image is done in the same way as during the initialization phase. As mentioned in 5.2, selected good points to track in the previous image are matched to the nearest point extracted in current image. Even if the point selection was done in 5.2, wrong matchings may occur due to some closely-spaced points. In this case, camera pose estimation by RANSAC will be done again.

7. Experimental Results

7.1. Settings

As mentioned in 3.4, we use real GIS data of the city of Nantes, France. In the database, 3760 intersections are included and processed to create indexes of point's IDs and their corresponding invariants as a pre-processing. As a query, a part of the map which includes 59 intersections is adopted. The number of buildings in the map is 4080.

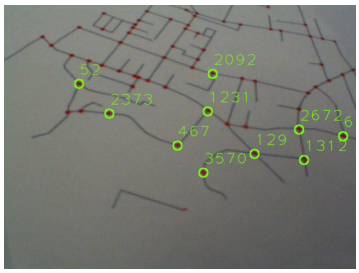
In our development environment, the laptop has Intel Core 2 Duo 2.2GHz and 3GB RAM. The lens distortion parameters and the focal length are calibrated beforehand. The camera by Point Grey Research has 640 × 480 size and is connected by Firewire.

7.2. Tracking Selection Result

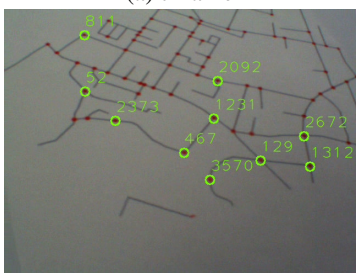
In 5.2, the method for selecting good points to track was described. The good points are updated at every frame in order to adjust to the change of viewpoint.

Figure 10 illustrates when a point is replaced with another better point to track in two continuous time-series images. In Figure 10(a), a point (ID:6) located near right image border was selected as a good point to track, but it wasn't any more in Figure 10(b). Since the distance between the point (ID:6) and the image lower border became too short in Figure.10(b), a new point (ID:811) located on left upper area was selected.

The good points are selected depending on the change of viewpoints. For this reason, continuous tracking is successfuly done.



(a) t frame



(b) t+1 frame

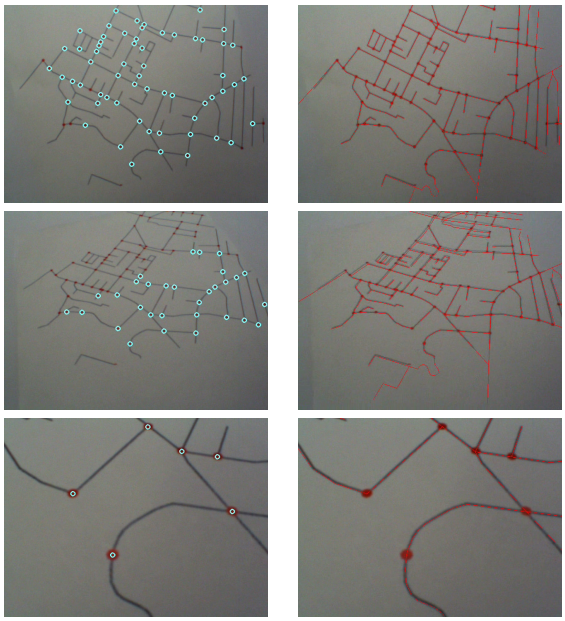
Figure 10. Tracking Selection Result

7.3. Re-projection Error of Roads

In this section, we represent the accuracy of the estimated homography in the tracking phase. In Figure 11, the first column shows matched points that are generated for calculating a more accurate homography in 5.1. The second colum shows re-projection of roads in GIS data.

In the first row, the input was captured from a position near to top view. The re-projection of roads is completely registered in the map of roads since the distribution of matched points was uniformly widespread in the image. In the second row, perspective distortion occurred and the matched points were dominated in the middle part of the image. For this reason, a slight re-projection error occurred in the upper part of the image. In the last row, there were only

six matched points. Since many points should be included in the image during the initialization phase, the initialization could not be done in this case. On the other hand, our tracking method works when the number of tracked points is more than five and tracking was successfully done in this case. The re-projection of roads was perfectly matched on the printed map's roads.



(a) Matched Points (b) Reprojection

Figure 11. Re-projection Error of Roads

7.4. AR Representation and its Computation Cost

After initialization of the camera pose, the camera can be moved to an arbitrary viewpoint. This section presents the AR representation of 3D GIS on the images captured from several viewpoints in Figure 12.

For achieving an AR representation system, we should take the computation time into account. Table 2 represents each computation time of all processing. Every computation time was calculated by averaging 100 images' results. The latest processing is the overlaying of 3D GIS on the image depending on the viewpoint.

LLAH + LLAH Selection's cost is a hundred times the Tracking Selection's one. The main difference between the initialization phase and the tracking phase is the number of points that should be included in an image. Since the Tracking Selection needs more than five points and the number of tracked points is always less than ten in our case, the computation cost is much lower than that of LLAH + LLAH Selection.

Table 2. Computation Costs

Process	Time (msec)
Point Extraction	42
LLAH + LLAH Selection	12
Homography by RANSAC	16
Tracking Selection	0.1
Image Capturing	19

8. Conclusion

In this paper, we have presented an AR representation system for 3D GIS that is based on the augmentation of a physical map including intersections. It provides a natural device for 3D GIS information representation and manipulation.

Our approach can be divided into two main processing steps. In the initialization step, LLAH-based intersection recognition was applied by using a top view image. After the initialization was done, point tracking with less than ten points was applied in order to track the camera pose even if the camera was close to the physical map. Experimental results show that overlaid 2D/3D GIS information could be seen naturally and our computation times were compatible with an AR representation system.

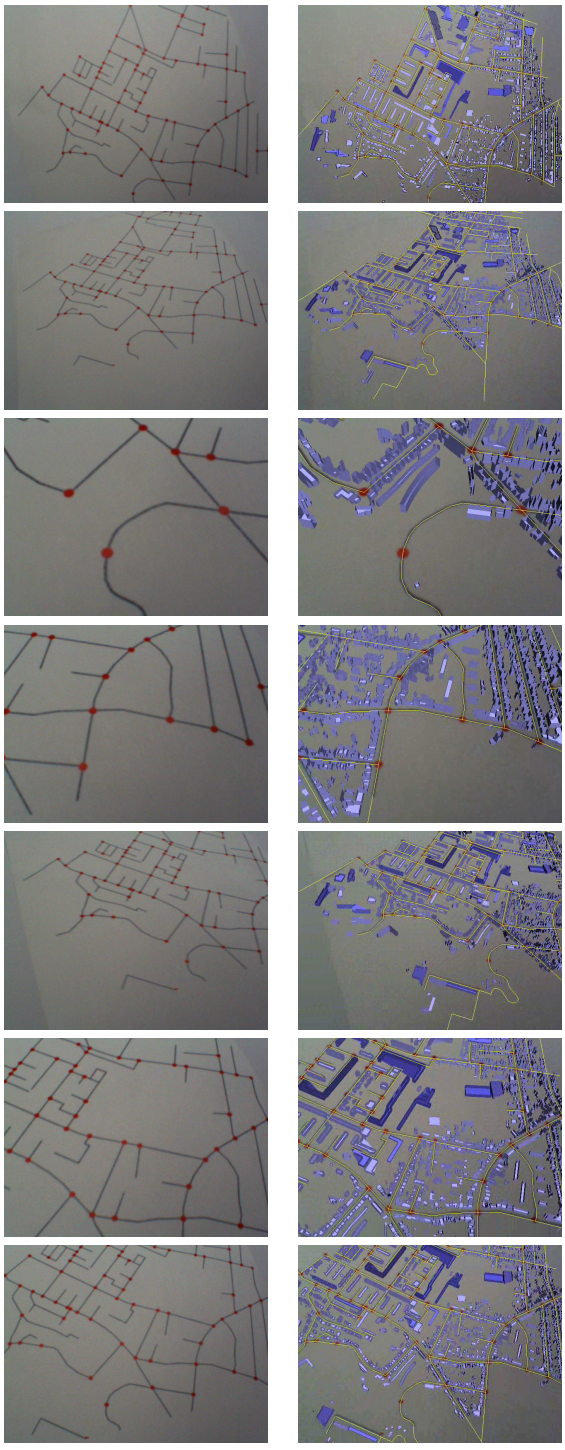
Our future work will be centered around three main topics. First, we will be using a real physical map, easier to manipulate, but requiring more image processing to recover the features needed in the initialization phase. Second, a map contains more information than just intersections, and this could be used to extract other features such as connectivity. Vectorization of roads from a satellite image will be helpful to extract roads and their connectivities from a physical map. Last, research works about data representation will be necessary to address both technical issues, for example the data size, 3D label placement, interaction in AR, and geographical issues like information semiology in 3D AR application and map generalization.

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.

References

[1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45:891–923, 1998. 2



(a) Inputs (b) Rendered Models

Figure 12. 3D GIS Rendering

[2] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *Proc. ECCV*, pages 404–417, 2006. 2

[3] E. Bocher, T. Leduc, G. Moreau, and F. G. Cortés. Gdms: An abstraction layer to enhance spatial data infrastructures usability. In *Agile 2008*, 2008. 4

[4] M. Datar, P. Indyk, N. Immorlica, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. SCG*, pages 253–262, 2004. 2

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981. 3

[6] N. R. Hedley, M. Billinghurst, L. Postner, R. May, and H. Kato. Explorations in the use of augmented reality for geographic visualization. *Teleoperators and Virtual Environments*, 11:119–133, 2002. 1

[7] M. Iwamura, T. Nakai, and K. Kise. Improvement of retrieval speed and required amount of memory for geometric hashing by combining local invariants. In *Proc. BMVC*, pages 1010–1019, 2007. 2, 3

[8] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proc. CVPR*, pages 506–513, 2004. 2

[9] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. ICCV*, pages 238–249, 1988. 2

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 2

[11] T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *LNCS (Proc. DAS'06)*, pages 541–552, 2006. 2, 3

[12] S. Neubauer and A. Zipf. Suggestions for extending the ogc styled layer descriptor (sld) specification into 3d - towards visualization rules for 3d city models. In *Urban Data Management Society Symposium*, Stuttgart, Germany, 2007. 4

[13] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, pages 2161–2168, 2006. 2

[14] R. Nussinov and H. J.Wolfson. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. In *Proc. Nat'l Acad. Sci. U. S. A*, pages 10495–10499, 1991. 2

[15] G. Reitmayr, E. Eade, and T. Drummond. Localisation and interaction for augmented maps. In *Proc. ISMAR*, pages 120–129, 2005. 1

[16] Y. Uematsu and H. Saito. Vision based registration for augmented reality using multi-planes in arbitrary position and pose by moving uncalibrated camera. In *Proc. MIRAGE*, pages 99–1019, 2005. 5

[17] H. J. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Comp. Sci. and Eng*, 4:10–21, 1997. 2