Construction of a 3D city map using EPI analysis and DP matching

Hiroshi KAWASAKI, Tomoyuki YATABE, Katsushi IKEUCHI, Masao SAKAUCHI
Institute of Industrial Science, University of Tokyo
7–22–1 Roppongi, Minato-ku, Tokyo, 106-8558, Japan
h-kawa@sak.iis.u-tokyo.ac.jp

ABSTRACT

In this paper, we propose an efficient method to make a 3D city map from real-world video data using digital map. To achieve this purpose, mainly two problems mentioned below exist.

1. Video data is usually huge, so suitable data structure is needed for an efficient handling
2. To construct 3D map from video data using digital map, matching between video data and digital map is needed

To solve the first problem, we propose an automatic organization method focusing on video objects to describe video data. Actually, we divide spatio-temporal space into video objects which represent individual buildings by EPI analysis.

To solve the second problem, we use DP matching method. When we apply DP matching method, we use depth information obtained from the real-world video by EPI analysis. And this method successfully improved the precision and reliability of the DP matching.

On its implementation, we made two sample applications to demonstrate an effectiveness of our proposed method. First application is a “Interactive Q&A System” which extracts an video object from video data automatically and makes DPmatching between video data and digital map. Second application is a “3D modeling program” which makes an VRML 3D map from DPmatching result.

Both applications prove that our method is working enough efficient for our purpose.

1 Introduction

Recent progress in technology makes possible the seamless integration of the virtual world and the real world, and as a result of this progress, mixed reality has become an important and popular technology. In this mixed reality, real-world data and computational data must be related to each other and much research has been done in this area. To integrate the real-world and the virtual world, photometric(image based) or geometric matching is necessary. However, most research is concentrated on one side and less addresses both. So we propose a new matching method between real-world video data and digital maps containing geometric data.

Of course we can not match the different kind of data, so first we must make the same kind of pattern from both real-world video and digital maps. In the following section, we explain how to make the same kind of pattern from real-world video and digital maps.

Consequently, we propose a matching method between the real-world and digital maps. It also contains a method to obtain 3D information by using the optical flow which is acquired from video-data to make good quality matching.

Finally, we show some applications of an example of this matching method and we also present acquired models of a 3D city map.

We use following specially taken video data for our research.

Target Video Video is recorded from a vehicle on which a camera is perpendicularly installed to the direction of movement.

2 Make Pattern from real-world video data

Video data usually contains no typical structure to make matching with a digital map. So we have to find out convenient structure for matching from video data. And video data is too large for reasonable handling on a computer so we have to develop some efficient management method.

To satisfy these conditions, we propose “panoramic boundary edge pattern” for matching and “video object” based data structure [6] for efficient handling of video data. Now Video object is popular because MPEG–4 systems adopt it for its main technology, but, extracting video objects from video data usually incurs some difficulties and still remains as a difficult matter. In this paper, we propose a specialized extracting method for real-world video which is specially recorded as mentioned above. This video object extracting method is based on EPI(epipolar plane image)[2]. Fig.1 is a usual EPI, but, in this situation, most buildings are the same distance from the vehicle, the EPI looks like Fig.2, and each belt is theoretically matched with one building. Furthermore, the EPI is based on the vehicle having a constant velocity, and since the vehicle can’t move of a fixed speed, some solution is needed.

Considering the facts mentioned above, we created the panoramic boundary edge pattern as using the following process.

1. make “edge based EPI” from video data
2. extract video object from video data using EPI analysis

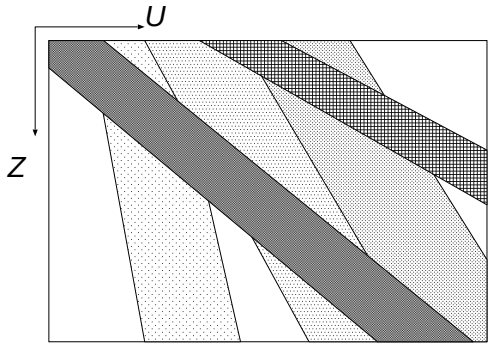

Figure 1. Tracking Image of singular point on EPI.

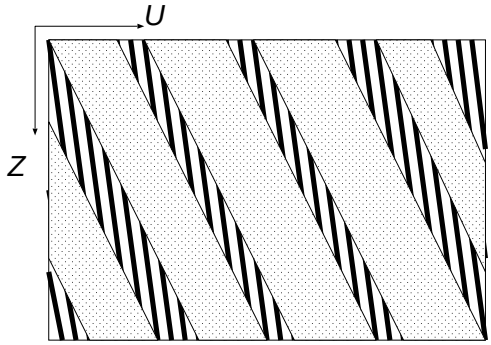

Figure 2. EPI made from the same depth of buildings.

3. make panoramic boundary edge pattern using video object database

Details of the process are explained in the following subsections.

2.1 Making of "edge based EPI"

There was a lot of research about EPI in the past, but most of them used the video carefully taken in the laboratory and not applied to real-world video data. Actually, real-world video has a lot of noise and difficult to simply apply EPI analysis to it. In this paper we propose a new EPI analysis method called "edge based EPI" to analyze real-world video data efficiently. In the following section, we show how to make the "edge EPI" and how to carry out "edge based EPI" analysis.

2.1.1 Edge detection using perceptual organization

Before we explain about edge detection techniques, we must discuss why we chose this technique to get the boundary of the building and why other effective techniques like texture based segmentation are left. It is mainly because real-world video of an urban city usually contains a lot of linear edges which commonly coincide with the structure of buildings. The detected edges exactly coincide with the boundary of the buildings in many cases.

The sequence of edge detection using perceptual organization [8]is as follows.(see Fig.3 as reference)

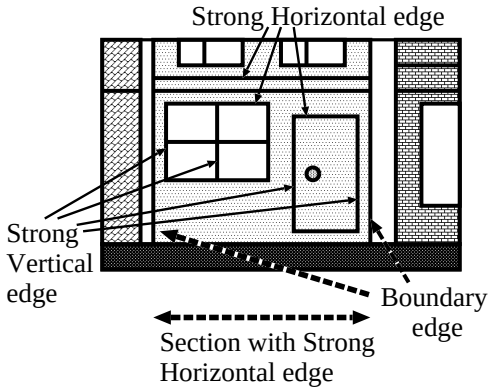

Figure 3. Structure of the building

1. Detect the vertical edge by using a Canny algorithm and if the edges are at an angle of $\frac{\pi}{16}$ or less on both sides, then grouping them onto one edge.
2. In the same way as Proc.1, detect the horizontal edges.
3. The vertical edges detected by Proc.1 have a lot of errors, such as window frames, etc. To solve this problem, remove all the vertical edges in the sections where strong horizontal edges are detected.

2.1.2 Assumption of velocity

First, we get the motion vector by a simple block-matching method. However, the simple block-matching method usually has a lot of noise, so we apply the Gaussian filter shown on (1) to assume a reasonable velocity.

$$G[i] = (2\pi\sigma)^{\frac{1}{2}} \cdot \exp(-\frac{1}{2} \cdot \frac{(velocity[i] - ave)}{\sigma}) \quad (1)$$

A sample of the retrieved data is shown in Fig.4. It is possible to see the broken lines(retrieved by simple block-matching method) have improved to a solid line(apply a Gaussian filter).

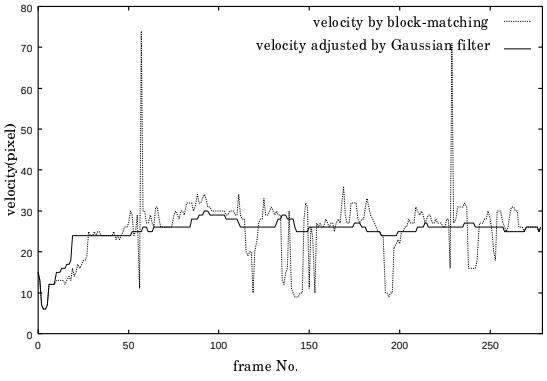

Figure 4. Result of assumed velocity

2.1.3 Result of our method

The EPI made by plotting the edges in Sec.**2.1.1** with the adjustment of velocity in Sec.**2.1.2** is shown in Fig.5(left be-

low) and Fig.6(above). This is what we call "edge based EPI" and this EPI only has dots on it, so we can easily detect lines on this image plane.

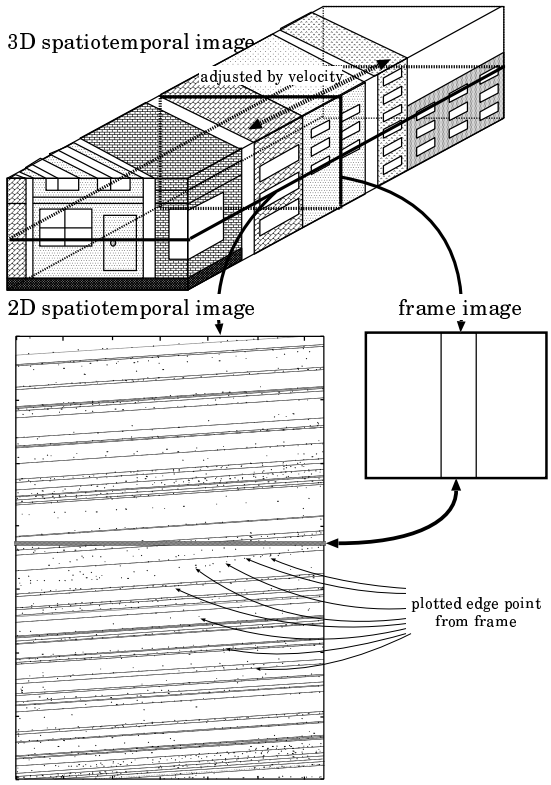

Figure 5. Spatio-temporal image consists of detected edge

2.2 Extracting video object using "edge based EPI"

In Fig.6, you can see plotted dots form many lines with a almost the same slope. To make a video object, we must get a straight line from the EPI to distinguish the belt from the other one. In this paper we use the Hough transformation shown as (2) to get a straight line from the EPI.

$$\rho = x\cos\theta + y\sin\theta \tag{2}$$

The slope in the EPI is almost the same, so the angle of slope is restricted to a narrow range, and therefore the calculation cost is not so great. Then we remove the lines concentrated in narrow space and leave only the ends of both sides. The result of these processes are shown in Fig.6 (below), and in the end, we can obtain a video object simply cut off the EPI on this estimated straight lines.

2.3 Panoramic boundary edge pattern

We can have PVI (Panoramic View Image) by cutting EPI vertically shown in Fig.7. We restore the video object data on PVI and remove all textures on PVI and leave only boundary edge of the buildings on it, then we can have the "panoramic boundary edge pattern" as shown in Fig.7(right below).

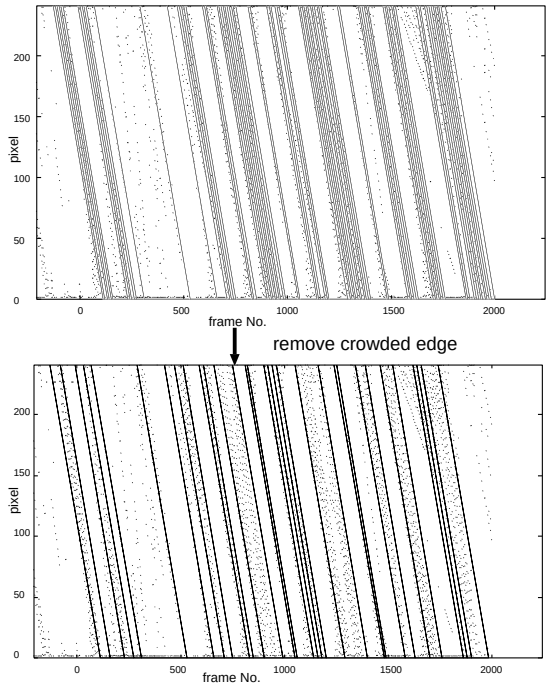

Figure 6. EPI made from edge data

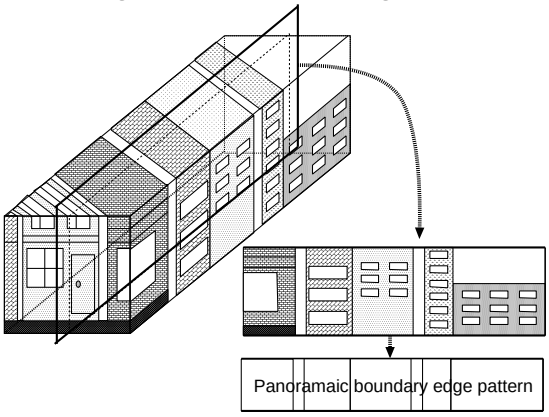

Figure 7. "Panoramic boundary edge pattern"

3 Make a pattern from a digital map

Since the target is a real-world video taken from a vehicle on the road, the model created from the digital map needs to correspond to this. By relating the geometric operation to the digital map, the pattern of the boundary of the building seen from the road can be obtained. The procedure for making a boundary pattern of the buildings along the road is as follows.

1. Describe at least two buildings which face to the route.
2. Determine the path which ties the two described buildings in Proc.1.
3. Create a model of the boundary patterns of the building which exists between the described buildings.

In Proc.1, both description done manually or automatically by GPS is supposed in our system. On this implemen-

tation manual description over network is supported. Details are explained in Sec.**6.1**.

In Proc.2, since the described information in Proc.1 is only the location of the two buildings, the system must assume the route which the vehicle travels. This assumption is done automatically as follows. Since the road data provided by the digital map is as short as $20 \sim 30m$, the system first searches the two roads which each building touches respectively. Then, the shortest path to connect the two acquired roads is obtained.

In Proc.3, the boundary pattern of the buildings is created by orthogonal projection from the route to the buildings (consisting of polygonal data) in the digital map.

Fig.8 shows the automatic modeling of the boundary pattern of the buildings obtained from the digital map using the two points described on the map.

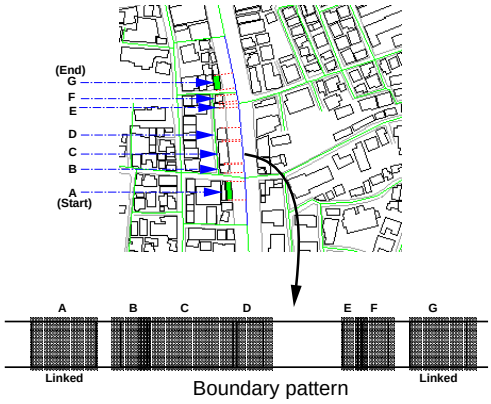

Figure 8. Automatic configuration of pattern.

4 Acquiring depth from video

We use DP matching for matching method. Scaling and positioning by DP matching allows us some errors in the modeling of the boundary pattern, but as the distance of the start and end point becomes greater, errors increase. Also, DP matching needs a pair of correspondent points at the both ends of sides, which usually contain some errors.

So far, we have implemented these matching methods using simple DP matching, but errors begin to be conspicuous when 6 or 7 buildings exist between the correspondent points and this method is not sufficient in practical use. In this paper, to avoid these errors, we use depth information in addition to simple edge pattern and achieve high quality matching.

In this section, we shall explain the reliable method of acquiring depth data from real-world video.

Regarding the acquisition of 3-dimensional (3D) information in a townscape, various research has been made in the past. Normally a camera is fixed in the direction of movement and analyzing the cross section of the spatiotemporal image of the EPI [7] is suitable for the urban city in which most structure consists of planes. Moreover, analysis using the factorization method [11] with the restriction that urban city's mostly consist of planes, such as buildings, etc. is considerably effective.

But, in an actual city environment, it is difficult to acquire 3D information as right theory due to many obstacles, such as telegraph poles or trees and buildings sometimes consist of complicated structures instead of plane surfaces. Actual buildings contain many complex texture and it often results in difficulties on extracting the feature points.

Overall, in this paper, we apply a technique called the "dynamic EPI" method together with the "panoramic boundary edge pattern" mentioned in Sec.**2.2** to acquire depth information for the matching.

To acquire depth information, we use the EPI plane made in Sec.**2.2**. On this EPI plane, boundary lines are already detected and the zone inserted into the adjacent boundary edges is considered one building or a gap between two buildings. So, acquisition of depth information is made by assuming the relationship of the adjacent zones by the motion vector analysis which represents the zone.

The actual process of acquiring depth is as follows(Fig.9).

1. All the frames are divided into vertical slits, and the motion vector of all those slits are assumed using a block-matching method.
2. Cluster the motion vectors of all the slits included in the same zone. Then select the maximum cluster in the zone and calculate the average of this maximum cluster and define this value as a representative value of this zone.
3. By using these representative values, assume the target zones depth information

This method is like a video game technique in which the front object moves faster and the background moves slowly. The depth information of the zone acquired by this technique is shown in Fig.10. The signs in this figure are the index of the building which are all done manually and the capital-letter J expresses the intersection.

It turns out that this method still produces some noise, but can acquire depth information to a satisfactory degree. In particular, deep depths such as intersections can be detected with high stability by setting up a good threshold. So, using the depth information for matching below, mainly intersection data is used.

5 Matching between the real-world and digital maps

In the past, there have been some attempts to match the real-world and maps. For example, using an aerial photograph [9] or silhouette of a distant view of buildings [5] as a real-world image . The former uses s stereo matching method with feature extraction and the latter uses a DP matching method. Since they use different kinds of images for the real-world, matching method is certainly different. In this paper we use the DP matching method for the following reasons.

- DP matching is an old and simple method and there is a lot of work to fasten the calculation. In addition, implementation is also easy.

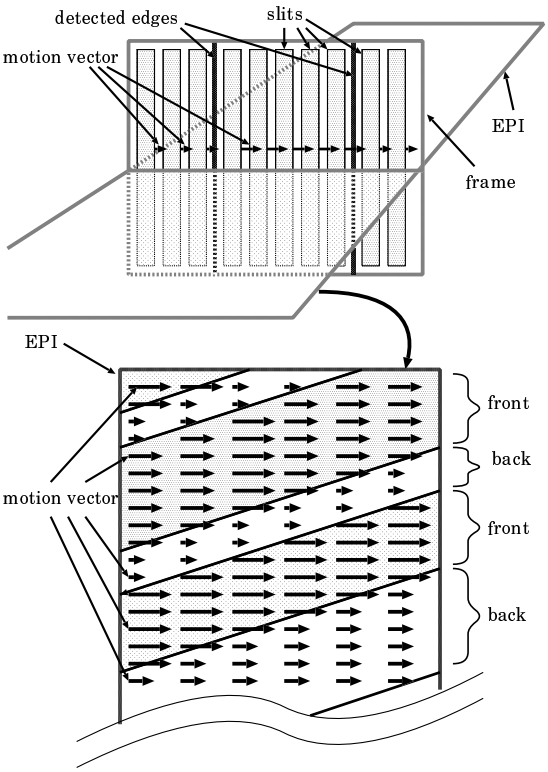

Figure 9. Acquire depth by motion vector

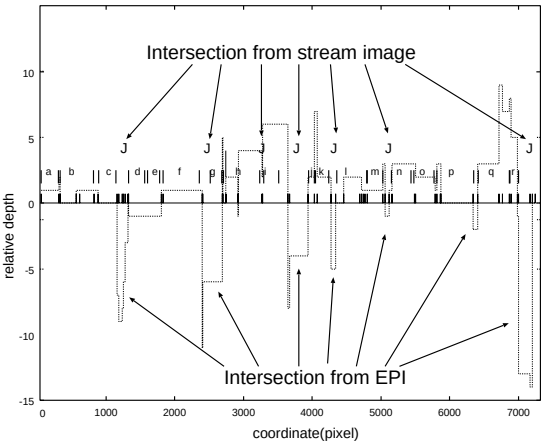

Figure 10. Estimated depth.

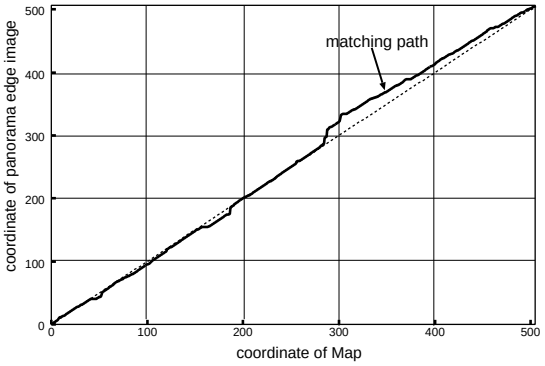

Figure 11. Path of DP matching.

- Matching pattern made in this paper usually changes its length, but not its order.
- We can easily add new pattern like depth pattern on original one.

For matching, we use the "panoramic boundary edge pattern" acquired from the real-world video stated in Sec.**2.2** and the boundary pattern models obtained from the digital map stated in Sec.**3**.

5.1 DP matching

When the two buildings are described in real-world video, the two buildings correspond to the buildings in a digital map. With this described correspondent point, we can make DP matching between video data and digital map.

The results of matching with depth information is shown in Figs.11 and 12.

Moreover, the matching result in the case of no depth information is shown in Fig.13.

5.2 Evaluation

Comparing Fig. 12 with 13, it turns out that the accuracy of matching is improved greatly by using the depth information obtained by the "dynamic EPI"analysis. Fig.14 shows the squared error values of both matching result by bar graph.

It is a well known fact that DP matching is greatly dependent on the matching weight, but in this implementation, several experiment shows weight doesn't influence too much. It is mainly because we can obtain intersection from video data with good quality.

Also, Effectiveness of edge detection is important factor for this method. We tried detecting the edge with various threshold and finally we obtain robust and reliable value irrespective of video data and map.

We also tried the same experiment in several streaming videos, and have almost the same and sufficient result.

6 Implementation of prototype systems

6.1 Real-time Query and Answer system

Fig.15 shows an example image of the system. This system has mainly two functions, one is to get corresponding point for DP matching and the other is to answer the queries asked by users. Both functions can be done over network, in particular the Internet, and be processed in real-time. To realize this function, the systems are all written in Java language and work on any browser which can run Java with JMF [1].

Fig.15 is a sample scene of describing corresponding point on streaming video by indicating building by pointer.

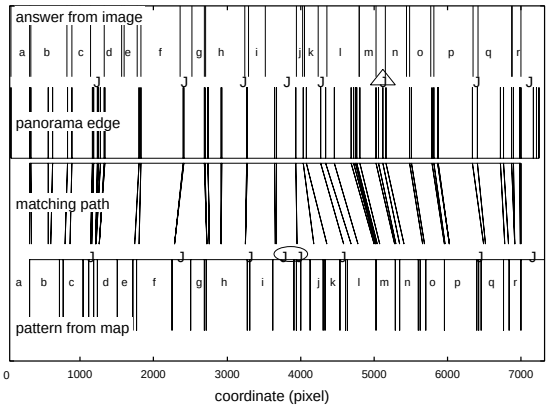

Figure 12. Result of DP matching.

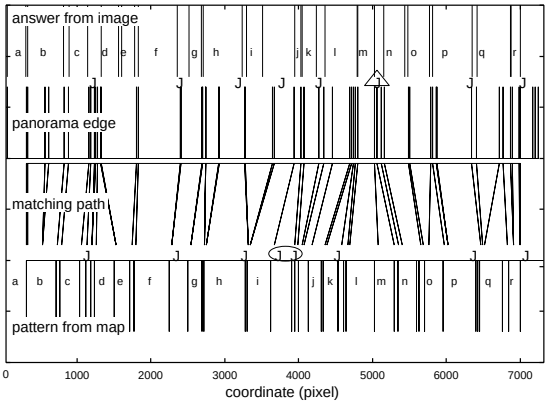

Figure 13. Result of DP matching without depth information.

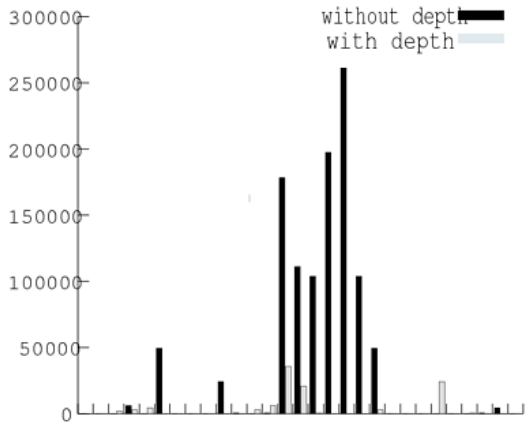

Figure 14. Error value of DP matching

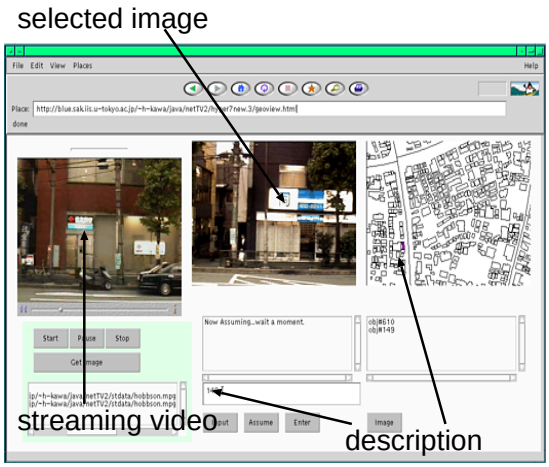

Figure 15. A snapshot of description

6.2 Retrieval of individual building images

This system creates a texture data base of buildings by cutting and dividing a texture image from the video data for each individual building.

All divisions are done automatically by projecting the corresponding edge pattern obtained by the DP matching to the panoramic image(Fig.16) made using a mosaicing method [10].

An example of the retrieved texture data base is shown in Fig.17.

6.3 An automatic construction of 3D virtual map

Using the results of the former two systems, we made a system which can generate a 3D virtual map automatically with a VRML form. This system works in two phases as follows.

1. make a geometric model using the digital map
2. put the texture data (retrieved in Sec.**6.2**) onto a geometric model

In both processes, sufficient accuracy of the matching between video data and digital map is needed for practical use.

A 3D map made by this system is shown in Fig.18.

7 Conclusions

We proposed a automatic 3D modeling method of a city map from real-world video using a digital map. To achieve this target, we must solve some difficult problems, such as structuring of video data and stable matching method. The former problems are settled by adopting video objects for data structure, and the latter problems are fixed by using depth information for matching. To acquire depth information from video data, we propose the “dynamic EPI” analysis. The “dynamic EPI” unlike usual EPI analysis which concentrated on static image processing uses motion vector. And by “dynamic EPI” analysis we successfully detect deep depth with robust and good quality.

On its implementation we divide the required functions to make 3D models of city maps into three systems. The first system is the Q&A system used to make important and basic matching between real-world video and the dig-

Figure 16. Panoramic image by video mosaicing.

Figure 17. Texture database of buildings.

ital map(Sec.**6.1**). The second is an automatic retrieval system which retrieves an individual texture of the building from video data(Sec.**6.2**). The third makes a 3D map by integrating both the first and second system's result into VRML(Sec.**6.3**).

In the future, decreasing errors caused by the many obstacles existent in the city, such as telegraph poles and trees is needed. We are now trying to acquire more precise depth data and boundary edges of buildings from video. Moreover, examination of the further effective use of the 3D model still remains an important theme.

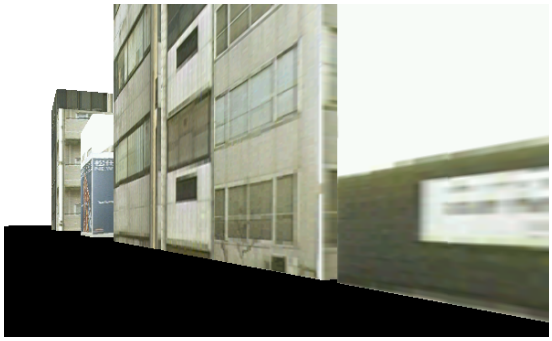

Figure 18. Virtual 3D map of VRML.

References

[1] Java(tm) media framework api home page. `http://java.sun.com/products/java-media/jmf/index.html`, Dec. 1998.

[2] R. Bolles, H. Baker, and D. Marimont. Epipolar plane image analysis:an approach to determining structure from motion. *Int.J.of Computer Vision*, 1:7–55, 1987.

[3] A. Hampapur, R. Jain, and T. Weymouth. Digital video segmentation. In *Proceedings of Second Annual ACM Multimedia Conference*, pages 357–364, Oct. 1994.

[4] H. Kawasaki, T. Yatabe, K. Ikeuchi, and M. Sakauchi. Automatic modeling of a 3d city map from real-world video. In *ACM Multimedia99*, Oct. 1999. (to appear).

[5] P. Liu, W. Wu, K. Ikeuchi, and M. Sakauchi. Recognition of urban scene using silhouette of buildings and city map database. *Proc. the 3rd ACCV*, 2:209–216, Jan. 1998.

[6] K. Manske, M. Muhlhauser, and S. Vogl. Obvi:hierarchical 3d video-browsing. In *Proceedings of Second Annual ACM Multimedia Conference*, pages 369–374, 1998.

[7] M. Notomi, S. Ozawa, and H. Zen. Modeling of urban scene by motion analysis. *IEICE Trans. Information and Systems*, J81-D-II:872–879, May 1998.

[8] S. Sarkar and K. L. Boyer. A computational structure for preattentive perceptual organization: Graphical enumeration and voting methods. *IEEE Trans.Systems, Man, and Cybernetics*, 24(2):246–267, Feb. 1994.

[9] Z. C. Shi and R. Shibasaki. Automated building extraction from digital stereo imagery. *Automatic Extraction of Man-Made Objects from Aerial And Space Images*, pages 119–128, May 1997.

[10] R. Szeliski. Video mosaics for virtual envrionment. *IEEE Comput.*, pages 22–30, 1996.

[11] C. Thomasi and T. Kanade. Shape and motion from image stream under orthography: A factorization method. *Int.J.of Computer Vision*, 9:137–189, 1992.