Spatio-Temporal Analysis of Omni Image

Hiroshi Kawasaki, Katsushi Ikeuchi, and Masao Sakauchi
Institute of Industrial Science, University of Tokyo
7–22–1 Roppongi, Minato-ku, Tokyo, 106-8558, Japan

Abstract

This paper describes an efficient method to obtain 3D information by using spatio-temporal analysis of omni images for outdoor navigation and map-making in the intelligent transportation system (ITS) application. Two types of omni-directional cameras are employed to make a spatio-temporal volume, which is a sequence of omni images stacked in the spatio-temporal space. For the spatio-temporal analysis of an omni image, we define several different cross sections in such spatio-temporal volumes, and examine characteristics of the traces of image features on the cross sections. We determine that the vertical straight lines in the real world are preserved as straight lines on these cross sections and that the degree of this slope represents the quotient of the velocity of the camera motion and the depth of the object.

To acquire 3D information using these characteristics, we propose a hybrid method of the epipolar-plane image (EPI) analysis and the models-based analysis. To demonstrate the effectiveness of this method, we present some experimental results and the ITS applications using an omni-directional video camera to obtain images in outdoor environments.

1 Introduction

Progress of the computational efficiency and expansion of the storage capacity has made it possible to analyze a huge amount of video data efficiently. As a result of these improvements, acquisition of 3D information from video data seems to be a realistic goal, a goal that many researchers have attempted to achieve. However, several difficulties still remain in applying these techniques to the real-world environment as opposed to using them in the experimental indoor laboratory environment.

This paper describes how we use an omni-directional camera to take video data and proposes a new method for acquiring 3D information from video data, a method that is more efficient and robust than previous methods. Section 2 describes several different kinds of cross sections in a spatio-temporal volume obtained by stacking acquired omni-images in the spatio-temporal space. Section 3 describes useful characteristics of traces of images features on the cross sections, and then presents an algorithm for acquiring 3D information from the previous analysis. Section 4 describes several experiments we performed to demonstrate the effectiveness of our method in acquiring 3D information as well as constructing the virtual city maps from those images. Section 5 concludes the paper.

1.1 Overview of Previous Work

In the past, various researchers have investigated the possibility of the acquisition of three dimensional (3D) information. Early attempts mostly employed stereo matching or the epipolar geometry method using a few images. Recently, however, the great increase in computational power and storage capacity has made it possible to analyze more images simultaneously; consequently, many methods have been developed. These methods are classified into two main types: the EPI analysis [11][2][1] and the factorization method [9].

The EPI method recovers depth information from known motion such as speed and route of the camera. In the actual analysis, it is usually assumed that the camera speed is constant and moving along a straight route. In addition, for usual EPI analysis, the object must have simple shape and plane textures.

In contrast, the factorization method has no restrictions on camera motion. This method uses a feature point and collects these feature coordinates into the measurement matrix to compute shape and motion simultaneously. This method seems very powerful and efficient in theory; however, it has a strong dependence on the accuracy of the feature tracking, accuracy which is sometimes quite difficult to achieve. And for realistic computation of the measurement matrix, the shape of the objects must consist of simple primitives.

After examining these two methods, we are determined to use EPI method because of two reasons. The first one is that it is often difficult to extract and track feature points reliably in the real-world scene, which is needed to exploit factorization method. This is mainly caused by noise, obstacles and complicated structures and complex textures of the object in the real-world. For the second reason, we assume that the target video is taken from the vehicle by an omni-directional camera which is mounted on the roof of the vehicle, and that this vehicle runs along the road through the city; these conditions satisfy the constraints of the EPI – constant speed and a straight route. Obviously, cities do not have only straight roads; however, we can assume that all curved roads consist of straight segments.

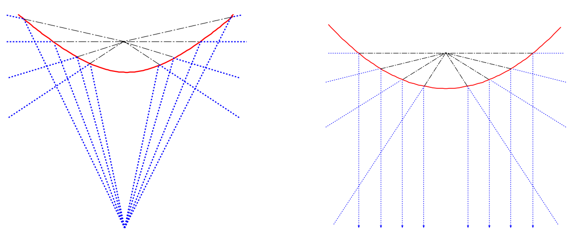
So far, there have been several attempts to extend or improve the EPI analysis[14], but because our spatio-temporal volume made from an omni-directional camera (STVO) is unlike a usual one, we cannot simply apply these methods, especially the shape of the slit. So, we try several shapes of the slit to make an image plane from STVO to find a good shape for EPI analysis. We also propose an efficient method which does not have to detect lines as accurately as is usually required for EPI analysis.

1.2 Omni-directional camera

The omni-directional camera is a camera designed to take an omni-directional view of the environment at the same time by taking the reflected view on a symmetrical mirror [7] [13] [6] [12] [10]. Many kinds of omni-directional cameras are now available. When we compare these omni-directional cameras, the structures are almost the same, but there are some significant differences in the shapes of the mirrors; these differences are the definitive characteristic of each omni-directional camera. Conical [10], spherical, hyperboloidal [12] and paraboloidal mirrors [7] are well known. Also, there are specially designed complicated mirrors[8]. In this paper, we select the hyperboloidal mirror and paraboloidal mirror for our analysis.

The omni-directional camera with a hyperboloidal mirror (Fig.1(a)) has two foci and makes it easy to reconstruct the perspective view. The hyperboloidal omni-directional camera, which can see a certain degree of downward direction, can take a panoramic view of its surroundings..

The omni-directional camera with a paraboloidal mirror (Fig.1(b)) has only one focus and can make an orthogonal projection image from the mirror and can easily reconstruct the perspective view. The paraboloidal omni-directional camera can take one hemisphere of the environmental view at a time.

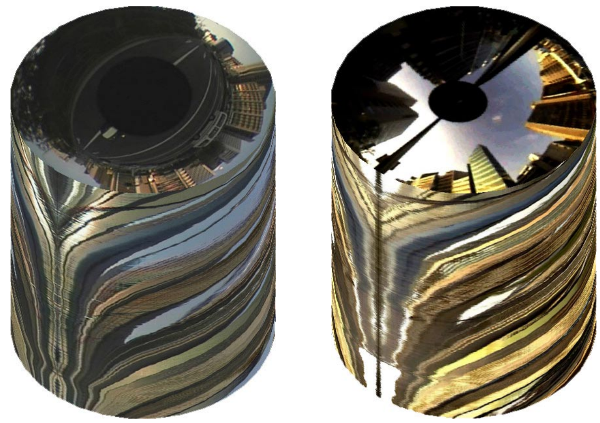


(a) (b)

Figure 1: (a)Hyperboloidal Omni-directional Camera (b)Paraboloidal Omni-directional Camera

2 Cross Section of Spatio-Temporal Volume

By accumulating omni images along the time axis, we can obtain a spatio-temporal volume of omni images (STVO) as shown in Fig.2 (a) and (b). For the analysis of this 3D volume, we can generate several different kinds of cross sections in this volume, and extract useful characteristics of traces of image features. In this section, we will discuss the relationship between the shape of the cross sections and the figure of traces which appear on these cross sections. For each frame image stacked in STVO, we call the direction in which the camera is moving along as "the vertical direction", while the direction which is vertical to the moving direction of the camera is "the horizontal direction".



(a) (b)

Figure 2: (a)Hyperboloidal omni spatio-temporal volume (b) Paraboloidal omni spatio-temporal volume

2.1 Stream line

In this subsection, we describe how we derive equations of a stream curve, a 2D projection of a 3D curve along which an image feature flows in the spatio-temporal space. This stream curve corresponds to an epipolar line in the traditional EPI analysis.

2.1.1 Hyperboloidal Omni-Camera

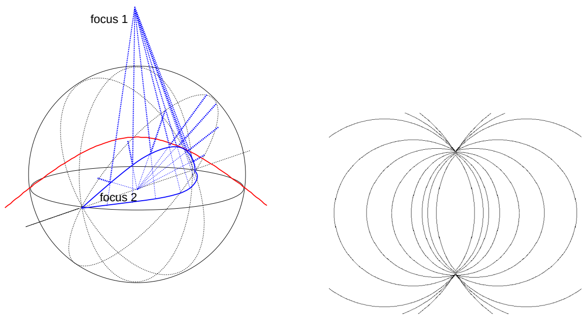
The equation of a solid of revolution, formed by rotating a hyperbolic curve, is represented as:

$$\frac{x^2 + y^2}{a^2} - \frac{z^2}{b^2} = -1 \tag{1}$$

A hyperboloidal omni-camera consists of a mirror, the shape of which is a hyperboloidal solid, and a TV camera. This hyperboloidal solid has two foci and the camera is placed at one of these foci, facing the mirror. In Fig.1(a), the lower point indicates the camera position and the upper hyperbolic curve depicts the mirror. Images taken by this camera, in which rays are depicted as dotted lines, are the same as the images taken by the another camera, located at another focus, with a center projection(Fig.3(a)), of which rays are depicted as broken lines.

By considering this reflection mechanism, we can obtain the equation of a stream curve from the equation of a hyperboloidal omni-camera system (1). As shown in Fig.3(b), the shape of a stream curve is an ellipse (zenith angle $|\theta| < arctan(\frac{b}{a})$) or a hyperbola (zenith angle $|\theta| \geq arctan(\frac{b}{a})$).

A stream curve starts from a focus of expansion (FOE) and ends at the vanishing point (VP). The center of this ellipse is located on the bisector of the diameter connecting the FOE with the VP. This center moves away from the diameter according to the zenith angle of image features. Here the north pole is defined as the direction of the optical axis of the camera.



(a) (b)

Figure 3: (a)model of hyperboloidal mirror (b)locus of horizontal line on omni-camera

2.1.2 Paraboloidal Omni-Camera

The paraboloidal omni-camera, the use of which is described in this paper, has a paraboloidal mirror, a solid of revolution generated from a parabolic curve. The equation of a paraboloidal mirror is represented as:

$$\frac{x^2 + y^2}{h^2} + \frac{2 \cdot z}{h} = 1 \tag{2}$$

Here, we use the same parameterization as [7]. As shown in Fig. 1(b), under this parameterization, all the incoming rays toward the focus of the paraboloidal mirror become parallel and orthogonal against the image plane of the camera after being reflected by the mirror.

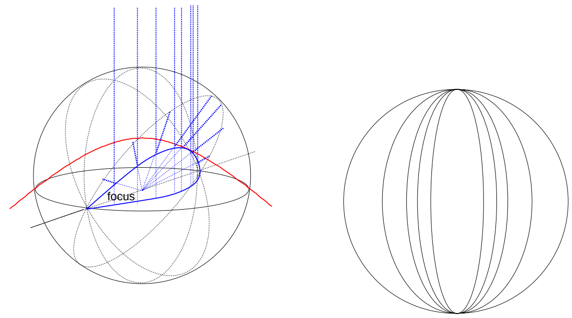
The shape of a stream curve is also an elliptic curve and its equation is expressed as:

$$\frac{x^2}{h^2 \sin(\theta)} + \frac{y^2}{h^2} = 1 \tag{3}$$

2.1.3 Examples

By cutting STVO, given by hyperboloidal and paraboloidal omni-directional cameras, along a stream curve, we can obtain cross sections, as shown in Fig.5(a) and 6(a), respectively. Here we denote those cross sections as hyperbolic and parabolic EPIs, respectively.

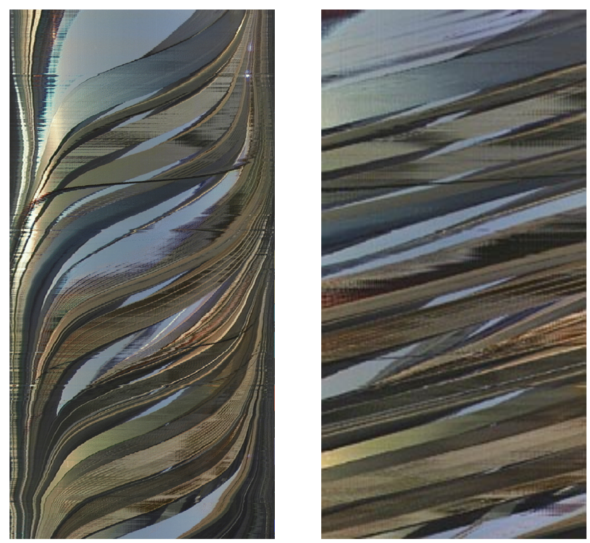
In the STVO, a far distant feature moves slowly, while a nearby feature moves rapidly. Due to the difference in the distance, a trace of an image feature, as



(a) (b)

Figure 4: (a)model of paraboloidal mirror (b)locus of horizontal line on omni-camera

shown in Fig.5(a) and 6(a), has a S curve on a cross section. By applying a perspective transformation, we can successfully transform this S curve into a straight line as shown in Fig.5(b) and 6(b) . These transformed EPIs are equivalent to those defined in [2].

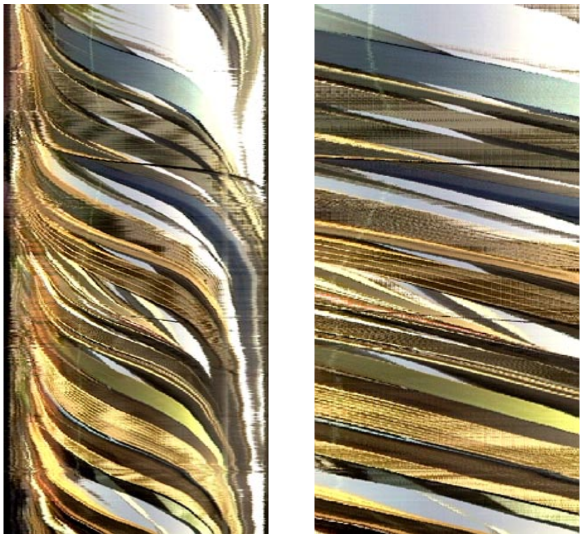


(a) (b)

Figure 5: Hyperbolic EPI

2.2 Radius line

A vertical line in the 3D space is projected as a straight line on an omni image, along a radius direction. In this section, we describe how we will cut a STVO along a radius line. This operation is equivalent to the one to extract a vertical slit of an usual perspective image and collect those vertical slits into an image. Thus, by applying this operation to the STVO, we can obtain a panoramic view image(PVI) defined by [13].



(a) (b)

Figure 6: Parabolic EPI

We can select any cutting line from infinite candidates of radius lines. All of these cutting lines make different PVIs. Usual PVIs, generated from perspective images, are restricted by the angle of the field of view of a camera. However, an omni-directional camera has a 360 degree field of view. Thus, by using a STVO, we can generate any PVI toward any viewing direction. This is the great advantage of STVO in retrieval of texture images and map-making.

We will show the example image of STVO cut along radius line in Fig.10(a) and three different PVIs given by three different radius lines (Fig.7, 8, 9). We can see the difference in these images depending on view directions.

Figure 7: PVI plane 5

Figure 8: PVI plane 19

Figure 9: PVI plane 39

2.3 Arbitrary straight line

The cross sections defined in Sec.2.1 and 2.2 have the same characteristics as those given by usual perspective images. Thus, we can apply the same analysis to these images. On the other hand, by cutting the STVO at an arbitrary straight line, we can obtain unique cross sections that cannot be obtained from usual perspective images, and have useful characteristics.

Before analyzing these unique cross sections, we will examine the relationship between arbitrary straight lines in omni images and vertical straight lines in the 3D world which draw radius lines in omni images. The vertical straight line located at (X_0, Y_0) in the 3D world draws a radius line on an omni image as:

$$y = \frac{Y_0}{X_0} \cdot x \tag{4}$$

We can express an arbitrary straight line as:

$$ax + by + c = 0 \tag{5}$$

We solve these simultaneous equations and arrive at the solution as:

$$x = \frac{c}{a + \frac{Y_0}{X_0} b} \tag{6}$$

$$y = \frac{c}{b + \frac{X_0}{Y_0} a} \tag{7}$$

We define the $u - t$ coordinate system, where the U axis is defined along the arbitrary straight line, and the T axis is the time axis. Then, the solutions are represented in this coordinate system as:

$$u = \frac{-c/b}{\frac{Y_0}{X_0}\cos(\theta) - \sin(\theta)}, \quad (\text{if} \quad b \neq 0) \tag{8}$$

$$u = -\frac{c}{a}\frac{Y_0}{X_0}, \quad (\text{if} \quad b = 0) \tag{9}$$

where $\theta = \arctan(-\frac{a}{b})$. And if the motion speed of the camera is constant (v) and the moving path is straight, we can describe Y_0 as :

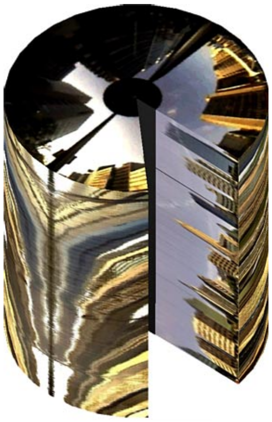
$$Y_0 = Y_{0original} - vt \tag{10}$$

We substitute this Y_0 to the equation (8),(9) and, as a result, we obtain the equation of the locus on $u - t$ plane as:

$$u = \frac{-c/b}{\frac{Y_{0original} - vt}{X_0}\cos(\theta) - \sin(\theta)} \quad (\text{if} \quad b \neq 0) \tag{11}$$

$$u = -\frac{c}{a}\frac{Y_{0original} - vt}{X_0}, \quad (\text{if} \quad b = 0) \tag{12}$$

Obtained equations in this section are effective in any omni-directional camera of which the mirror has a solid of revolution. Accordingly, we can apply these equations to both the hyperboloidal and the paraboloidal omni-directional camera. In the following section, we use two typical types of straight lines, horizontal and vertical lines, to form cross sections.



(a) (b)

Figure 10: (a)Radius cross section and (b)vertical cross section of spatio-temporal volume of omni-directional camera

2.3.1 Horizontal straight line

To make a cross section by using a horizontal straight line slit, we simply insert $a = 0$ (which means $\theta = 0$) into the equation (11) and we get:

$$u = \frac{-(b/c)X_0}{Y_{0original} - vt} \tag{13}$$

This equation describes a hyperbolic line on the $u - t$ plane. Fig.11 and 12 show the actual cross section by a real-world video; we can see that all buildings are distorted to the hyperbolic direction as theoretically derived.

Figure 11: Hyperboloidal omni-directional camera : horizontal cross section

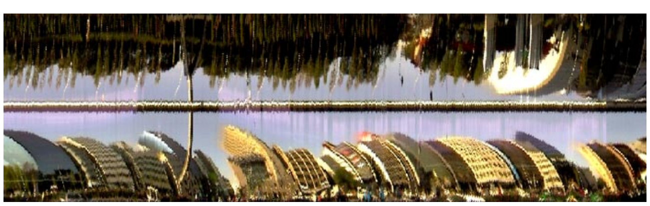


Figure 12: Paraboloidal omni-directional camera : horizontal cross section

2.3.2 Vertical straight line

A vertical straight line can be described as $ax = c$, where we substitute $b = 0$ for the equation (5). So we obtain the following equation from the equation (12).

$$u = -\frac{c}{a}\frac{Y_{0original}}{X_0} + \frac{c}{a}\frac{v}{X_0}t \tag{14}$$

This equation shows that the vertical edge in the real world describes a straight line on the $u - t$ plane when we cut STVO at a vertical straight line. Fig.10(b) shows the example image of the STVO cut along vertical straight line and Fig.13 and 14 show the actual cross section made by a real-world video. We can see that all edges and boundaries of buildings are preserved in a straight line on the cross section. We can also obtain 3D information from the directions of the line in this vertical cross section.

Figure 13: Hyperboloidal omni-directional camera : vertical cross section

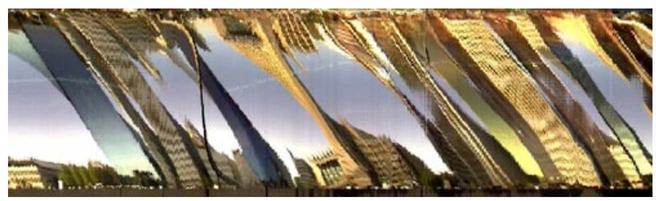


Figure 14: Paraboloidal omni-directional camera : vertical cross section

3 Retrieval of 3D information

In this section,we propose efficient method for obtaining 3D information from a cross section. First we will explain about the EPI-based analysis and then about the model-based one which is based on the method proposed in [5].

3.1 EPI based analysis

Due to the equation (14), vertical lines in the world coordinate describe straight lines on the vertical cross section and the coefficient of these straight lines represents the depth of the object and the velocity of camera motion. So if we know the velocity of the camera motion, we can determine the depth of the object. This is essentially the same as an ordinary EPI analysis; thus, we can apply most of technique developed for EPI analysis, especially the line detection technique.

On the other hand, several different characteristics exist between vertical cross section analysis and ordinary EPI analysis. For example, the analysis in the previous section is independent of the Z coordinate. However, since a 3D object has a finite vertical height, some straight lines disappear in the top and bottom of the image in Fig. 13.

3.2 Model based analysis

The EPI analysis is sometimes very difficult to apply to real world data, because keeping the camera speed constant and moving along the route in a straight direction is difficult in reality. Also, objects in the real world have complicated shapes and textures.

On the other hand, we can obtain models for most objects in the real world, especially for human work. And in the actual situation, when we use the video data taken from the vehicle as it travels along the street, we can use map data as a model.

Considering these situations, we propose a hybrid method using the EPI and the model-based analysis to avoid the difficulties of image processing caused by real world complexity. To put this analysis concretely, we perform matching between video data and models, and then retrieve 3D information from video data using the matching results.

3.2.1 EPI - EPI matching

Since models and video data are different representations, it is impossible to perform direct matching between them. We have to define some common basis for both sets of data. As stated in sec.2.3.2, we can easily make EPI (exactly cross section) with straight lines from omni-directional video data. We can also make EPI from models by simulating the camera motion and parameters(Fig. 15). Thus, we will employ this EPI as the common basis for matching.

There are also some advantages in using EPI for matching compared with the usual EPI analysis to extract 3D features from EPIs. Namely, the route does not have to be exactly straight and the camera motion does not have to strictly keep the velocity constant, because this EPI-based matching compares regions generated from both real and simulated EPIs. This method does not need precise line detection, because we are not going to extract parameters of lines; rather we will compare regions bounded by these lines for matching.

Thus, 3D information retrieval problem results in a 2D matching problem. By assuming that the order of objects does not change, we can use the DP matching. Next we will show actual method for DP matching adapted for the 2D pattern.

3.2.2 2D DP matching

We must carry out matching between the EPI made from the omni-directional video camera and the EPI made from a model. EPI is a 2D image, so we have to carry out matching between 2D images. In this paper, we propose a 2D image matching method using the DP matching; the actual process of this method is as follows.

1. Make a rectangular parallelepiped from two EPIs(Fig.16)

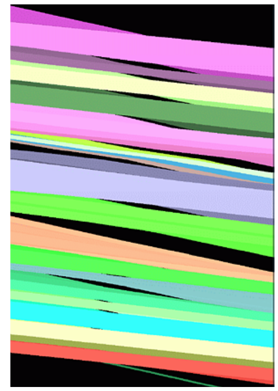


Figure 15: Model EPI made from digital map

2. Cut this 3D polygon by a horizontal plane and make a 1D search plane

3. Carry out 1D DP matching (usual DP matching) for each 1D search plane(Fig.17)

4. Iterate **1** to **3** using the constraints so that the matching path makes a continuous plane

4 Experiments

We have performed some experiments to demonstrate the effectiveness of our method. In the actual real-world scene, the EPI-based method mentioned in Sec.**3.1** still has some difficulties in detecting lines with enough accuracy and robustness to obtain depth information with error tolerance. However, the hybrid method mentioned in Sec.**3.2** works well in the real-world. We will show some experimental results in the following subsections.

Target omni-directional video data we use in this experiment is taken from a vehicle which runs along a city street.

4.1 Outline of actual STVO analysis

To obtain 3D information from omni-directional video data, we apply a hybrid model-based EPI method at this time, using a 2D digital map as a model. The actual process of acquiring 3D information is as follows.

1. Make the EPI from the video data.

   Simply cut the STVO at a vertical line and we can have one easily as stated in Sec. **2.3.2**

2. Make a model EPI from 2D map data.

   To make the model EPI, we make the virtual 3D city map in the computer by using digital 2D map data and simulating the motion and the camera parameters of the virtual camera.

3. Perform matching between two EPIs

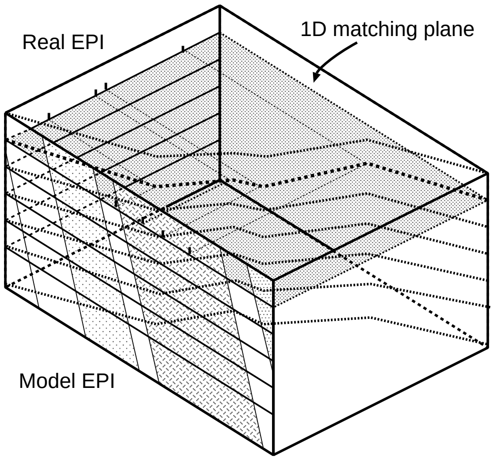


Figure 16: Concept of 2D DP matching

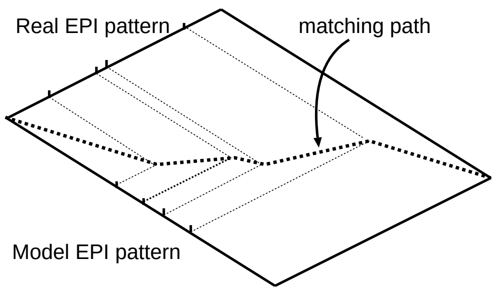


Figure 17: 1D search plane of 1D DP matching

Carrying out the DP matching, requires that we give the corresponding point. In this experiment we give the corresponding point as both start and end manually.

4. Restore 3D information to the video data

4.2 Result of 2D DP matching

To achieve high accuracy and robustness of the matching, we use three different patterns for matching, such as the edge of the building, the boundary of the building and the sky pattern. Fig.18, Fig.19(a) and (b) shows the result of DP matching between the EPI from the omni-directional video data and the model. When we see the Fig.18, we can identify that all matching paths make a smooth curved surface because of the constraint that the all matching paths are continuous to any direction. Fig.19(a) and (b) shows the selected matching path from all of the result paths in Fig.18; the upper one (Fig.19 (a)) is the matching result with only the edge pattern on the EPI and the lower (Fig.19 (b)) is the matching result with three patterns mentioned above.

Comparing (a) and (b) in Fig.19, we can identify that the accuracy of the matching result is greatly improved by increasing the patterns for the matching; this property is based on the feature of DP matching.

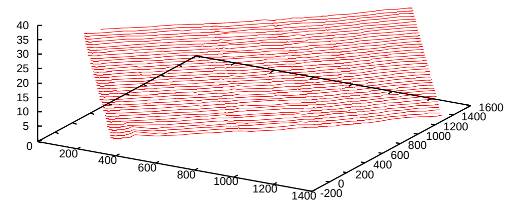


Figure 18: Result of the 2D DP matching

4.3 Applications

Importance of the collaboration between the real and virtual worlds is increasing. This technology is known as Mixed Reality(MR). 3D information collated with the video data can be efficiently brought into the MR[4]. We will show a sample application for the MR, which is effectively achieved by using our 3D information retrieval method. Fig.20 shows the sample snapshot of a virtual city and we can see that the texture of the side view is successfully mapped on the polygon of buildings.

4.4 Resolution

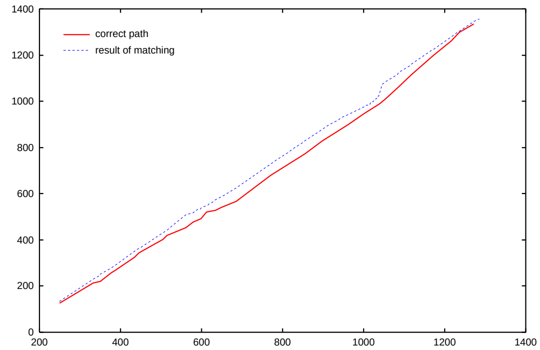
It is usually difficult to acquire high-resolution images taken by the omni-directional camera, especially after perspective transformation. High-resolution images are not necessarily needed for practical use of the application we mentioned in the previous section so, in this paper, we simply use acquired texture images to construct a virtual 3D city map. Of course, higher resolution images are more preferable for immersive MR systems, so we are now trying several approaches for it. (i.e., using high-resolution CCD and super-resolution algorithm.)

5 Conclusion

We made several cross sections from the spatio-temporal volume of omni images (STVO) and made analyses. We stated some characteristics for each plane, especially for a vertical straight line which made quite a unique and useful cross section for depth estimation. This vertical cross section preserves the vertical straight lines in the real world as straight lines on the cross section; the degree of this slope represents the depth of the object and the velocity of the camera motion.

We can estimate the depth with analysis of this vertical cross section. We proposed an efficient and robust method to obtain 3D information using the result of our analysis of the vertical cross section. This method is based on EPI analysis; we also used models to make the analysis realistic. To improve the result, we also proposed an efficient matching algorithm between EPIs, that is, 2D DP matching.

To demonstrate the effectiveness of our proposed method, we conducted several experiments using real world omni-directional video data and a digital map as a model. The result of the experiments show the effectiveness of our proposed method and matching algorithm. We also applied these results to Mixed Reality (MR) systems to show further successful utilization of our method.



Result of the DP matching (a)

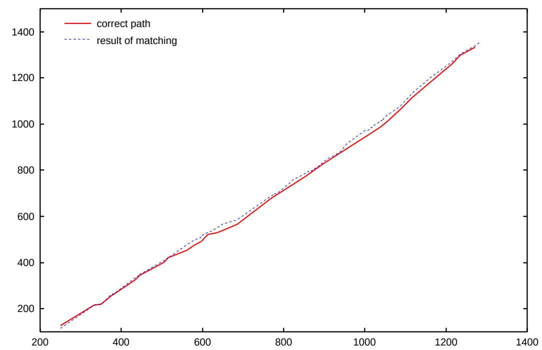


Figure 19: Result of the DP matching (b). Vertical axis represents the coordinate of EPI from the video data and horizontal axis represents the coordinate of the map data. If the model data is precisely corresponding to the real world, this matching path describes a straight line from the origin of the coordinate axes to the end point. But there are some errors in the map data, and the EPI made from video also contains some distortion errors. Thus, the actual path does not describe a straight line; rather, it describes the line as a solid line in the figures. A broken line is the matching result obtained by our matching method.

Acknowledgments

This work is supported in part by The Grant-in-Aid for Creative Basic Research #09NP1401 by the Ministry of Education, Science, Sports and Culture.

References

[1] H. Baker and R. Bolles. Generalizing epipolar plane image analysis on the spatiotemporal surface. *Int.J.of Computer Vision*, 3:33–49, 1989.

[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] T. E. Boult. Remote reality demonstration. In *Computer Vision and Pattern Recognition, Technical Demonstration*, pages 966–966, June 1998.

[4] H. Kawasaki, T. Yatabe, K. Ikeuchi, and M. Sakauchi. Automatic modeling of a 3D city map from real-world video. In *ACM Multimedia*, Oct. 1999.

[5] H. Kawasaki, T. Yatabe, K. Ikeuchi, and M. Sakauchi. Construction of a 3D city map using EPI analysis and DP matching. In *Asian Conference on Computer Vision*, volume 2, pages 1149–1155, Jan. 2000.

[6] Y. Onoue, K. Yamasawa, H. Takemura, and N. Yokoya. Telepresence by real-time view-dependent image generation from omnidirectional video streams. *Computer Vision and Image Understanding*, 71(2):154–165, Aug. 1998.

[7] N. K. Shree. Catadioptric omnidirectional video camera. In *Computer Vision and Pattern Recognition*, pages 482–488, June 1997.

[8] N. K. Shree and V. Peri. Folded catadioptric cameras. In *Computer Vision and Pattern Recognition*, volume 2, pages 217–223, June 1999.

[9] 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.

[10] Y. Yagi and S. Kawato. Panorama scene analysis with conic projection. In *Proc. Int. Conf. on Intelligent Robots and Systems*, pages 181–187, July 1990.

[11] M. Yamamoto. Determining three-dimensional structure from image sequences given by horizontal and vertical moving camera. *(D)*, J-69(11):1631–1638, Nov. 1986.

[12] K. Yamasawa, Y. Yagi, and M. Yachida. New real-time omnidirectional image sensor with hyperboloidal mirror. In *Proc. 8th Scandinavian Conf. on Image Analysis*, pages 1381–1387, May 1993.

[13] J. Y. Zheng and S. Tsuji. Panoramic representation of scenes for route understanding. In *International Conference on Pattern Recognition*, pages 161–167, June 1990.

[14] Z. Zhu, G. Xu, and X. Lin. Panoramic EPI generation and analysis of video from a moving platform with vibration. In *Computer Vision and Pattern Recognition99*, pages 531–537, 1999.

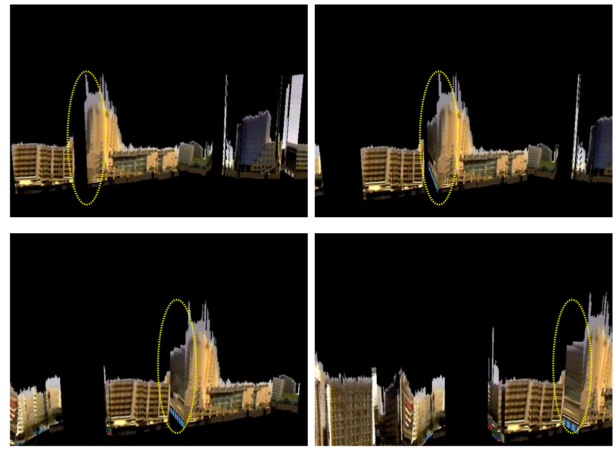


Figure 20: Snap shot of Virtual 3D map