Expanding Possible view point of Virtual Environment Using Panoramic Images

Takuji Takahashi Hiroshi Kawasaki Katsushi Ikeuchi Masao Sakauchi
Institute of Industrial Science
University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo JAPAN 106
E-mail : takuji@cvl.iis.u-tokyo.ac.jp

Abstract

This paper presents a new method for creating a 3D virtual broad city environment with walk-through systems based on Image-Based Rendering(IBR). In that virtual city, people can move rather freely and look at arbitrary views. The strength of our method is that we are able to easily render any view from an arbitrary point to an arbitrary direction on the ground in a virtual environment; previous methods, on the other hand, have strong restraints concerning their re-constructable areas.

One of the other applications of our method is a driving simulator in the ITS domain. We can generate any view on any lane on the road from images taken by running along just one lane. Our method first captures panoramic images running along a straight line, indexing the capturing position of each image. The rendering process consists of selecting some suitable slits divided vertically from stored images, and reassembling them to create an image from a novel observation point.

1 Introduction

The computer graphics community has expended much effort to creating large-scale virtual environments from real scenes. Generally speaking, 3-D geometrical models and surface attributes of the objects in the environment are used to create virtual environments, for example, a town or city. This method is called Image-Based Modeling (IBM). However, the ability of IBM is limited when it comes to creating images of relatively complex or small objects such as trees in city scenes; IBM also requires a huge amount of polygons for preparation.

Generating a 3-D virtual world directly from real scene images without using an explicit 3-D model and surface attributes of the objects is a promising technique. This method, referred to as Image-Based Rendering (IBR), creates new views by re-sampling those prerecorded pixels in a timely manner. This method can generate a highly photo-realistic virtual world.

"Aspen Movie Map"[6] was the pioneering work of this IBR technology. This system consists of a computer-controlled laser disc, which records the images along streets in the town of Aspen, CO.. The user can walk along the street where images have been captured. However, in this environment, the user can view the images only from the original viewpoint of the camera. In "QuickTime VR"[7] a series of captured environment images pasted on a cylindrical environment can generate a virtual world in which users can look around a scene from fixed points. This system, however, does not allow users to walk around in the environment.

One of the key concepts developed in the IBR is the plenoptic function. Three approaches have been described, that of "Lumigraph"[11] and "Lightfield"[10] and "Ray-space method"[12]. All of these approaches use a clever 4D parameterization of viewing position and direction. "Rendering with Concentric Mosaics"[13] is the 3D plenoptic function which, as its name says, creates concentric mosaics. In spite of the merits of the previously proposed plenoptic functions, the primary disadvantage of these plenoptic functions is that the range of field of view using these approaches is relatively small. It is not clear if these methods are feasible for viewing a wide real scene. One of the works most closely related to ours, rendering large-scale scenes, is Hirose and MR Systems's[14]. Using a vehicle-mounted image capturing system, they constructed a photo-realistic virtual world; in that virtual environment, the users had the impression of actually walking in the environment. However, this method can show only images captured near a path.

This paper proposes a new plenoptic function resolving those issues. Namely, the strength of our method is that we can create any view from any position to an any direction on the ground, in a wide area of a city. Thus, when synthesizing a large-scale virtual environment such as a city, our method has a great advantage.

2 Rendering Arbitrary View

2.1 Capturing Panoramic Images

For preparation, we store a sequence of panoramic images with a record of the capturing position of each image. One application of our system is creating a virtual whole city; another application is to build a driving simulator for which it is natural and simple for us to run on a road to capture prerecorded images of a city. Here, we are mainly concerned with the case where a camera runs along a straight line. As shown in Fig.1, by moving from C_0 to C_n, we capture images along with recording their positional information given by GPS sensor. Here we denote the ground plane as the $x-y$ plane and a panoramic image captured at (x_i, y_j) as C_k (x_i, y_j). We can pick up arbitrary slits from each panoramic image for reconstruction purposes.

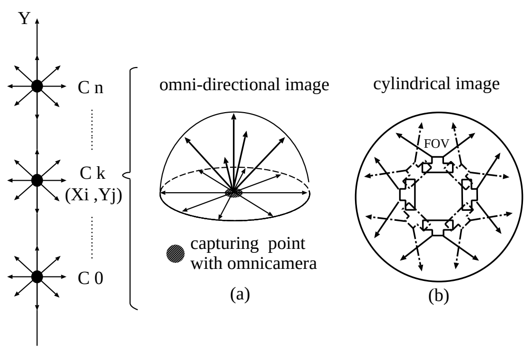


Figure 1: Capturing panoramic images

At each location, we construct a panoramic image. The simplest and easiest method for capturing panoramic images is to use an omni-directional camera. This type of camera has an orthographic lens that has a single effective viewpoint (see Fig.1-(a)). From the sensed omni-directional image, we can generate pure perspective images, and, thus can make panoramic images from the omni-directional image. Using one omni-camera, we can take images of 360 degrees in a horizontal direction; those images cover the northern hemisphere of a viewing sphere. Another method for capturing panoramic images is to arrange some cameras cylindrically as shown in Fig.1-(b). These cameras' optical axes intersect at one point with rays around the center of cameras. Projecting these perspective images to cylindrical coordinates, we store cylindrical images at each location. Hereafter, 'panoramic image' means both of these images.

2.2 Reconstruction of Novel Views

Given a series of panoramic images on a straight line, we can construct a novel view from any arbitrary region. Consider the case of rendering the novel image at the point P, as illustrated in Fig.2. For constructing the view from P, we need rays around P from R_s to R_e as shown in the figure. By finding the slits corresponding to the rays from stored panoramic images and then collecting those slits, we can synthesize a new view. For example, the ray of R_s is substituted for the θ_i ray in the $C1$ panoramic image.

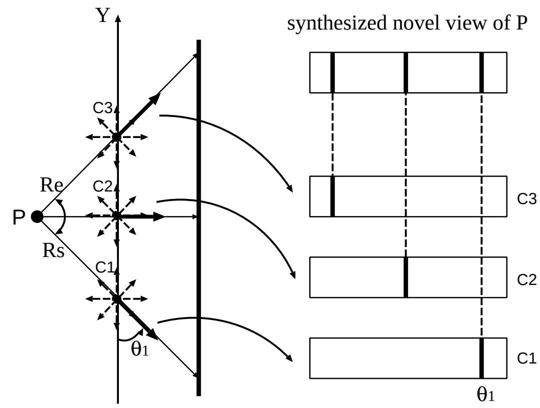


Figure 2: Reconstructing a novel view

3 Features

3.1 Region of Re-construct

In this section, we discuss the areas where it is possible to reconstruct novel views. First, consider an $x-y$ plane on the ground. Let us assume that the camera moves in a straight line for the interval from the position,C_0 to the position,C_n, see Fig.3. On or about the running line, we can render any rotation direction images. On the other hand, as the location of the viewing point is further away from the camera running line, the limit of rotation angle is more restricted. In the Fig.3, the novel images from point P can be rendered in the range as within arrows shown in the figure. Finally, upon rendering the view from the position P, we can render a novel view at only one rotation angle. Although there is a difference concerning the flexibility of the rotation angle, the re-constructable region on the plane is combining two sectors as FOV is the angle at the circumference. The area is shaded shown in Fig.3.

More importantly, we can enlarge the shaded area of the re-constructable region. By running for a longer distance, the boundary curve of the region is enlarged, i.e., the radius of the curve is lengthened. Additionally, rendering a view from the same point, the flexibility of rotation angle is increased. So, running in a straight line for an infinitely long distance, we can render a novel view of any angle and at any point on the ground plane.

3.2 Singular Direction

With respect to reconstruction, we can classify a novel view into two cases: with or without a singular direction. Here we define the singular direction along the direction

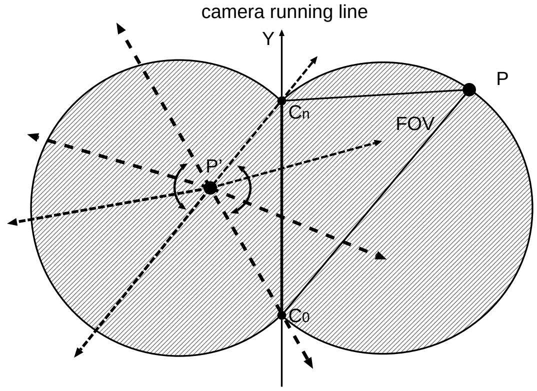


Figure 3: Reconstructable area on the ground

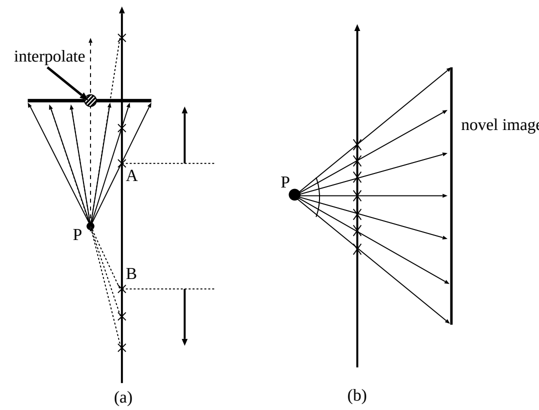


Figure 4: (a):With a singular direction (b):Without a singular direction

parallel to the running direction. Namely, the first case includes the ray parallel to the singular direction, while the second case does not. When reconstructing a view toward the moving direction from the driver's seat, we have to consider the first case, while for a side view, we consider the second case.

First, we will discuss the case with a singular direction. As shown in Fig.4-(a), a ray parallel along the singular direction does not exist except moving for an infinite distance. We have to interpolate this ray by using a morphing or other technique. However, usually along the singular direction, no object or very distant objects exist in the ITS applications. Thus, the distortion of any far objects is relatively small, or we could substitute this ray for the singular direction ray of stored images.

In the none singular case, a view to be rendered does not include the ray parallel to the singular direction. As shown Fig.4-(b), in this case, all necessary rays are contained in the series of panoramic images.

3.3 Vertical Distortion

Just as with the concentric mosaics, this system has the effect of the vertical distortion. We can devise several methods to reduce this distortion. If we know the distances between the camera optical center and the points in the scene, full perspective correction based on distances can be done. This method, however, requires acknowledge of geometry in the real scenes.

In many real scenes in the ITS application domain, it is a reasonable to approximate that pixels along a vertical line have the same depth such as walls of buildings. To estimate the depth value of each vertical lines, we employ the method called the "dynamic EPI" analysis [15]. Usually EPI analysis is done by static image analysis, but this method use the motion vector on the EPI plane. And as a result of this method, we can retrieve the depth value robustly and easily. Using this estimated value, we can scale the whole line uniformly.

4 Experimental Results

We have implemented this system and created many novel views from captured images in real scenes, both of an indoor area and a landscape of the city.

First, we describe experimental results for indoor scenes. Using a HyperOmni camera[16], and moving along a straight line, we captured a sequence of omni-directional images of our laboratory scene. From this camera's characteristics, we can easily generate a perspective image from omni-directional image.

Using these prerecorded images, novel views from position A, B, and C, as depicted in Fig.5, are rendered. And Fig.6$-(a)$, Fig.6$-(b)$, and Fig.6$-(c)$ are viewed from position A, B, C, respectively. These positions are depicted in Fig.5, close to the blackboard and the bookshelf in order. Note that the relation between the left edge of the blackboard and the right edge of the bookshelf behind the blackboard differs in those images. More precisely, in Fig.6$-(a)$, the right edge of the bookshelf and the left edge of the blackboard are not overlapped; in Fig.6$-(b)$, where the view point is closer to the objects than that in Fig.6$-(a)$, the left edge of the blackboard is lapped over the right edge of the bookshelf. In the Fig.6$-(c)$, the right edge of the bookshelf is completely occluded by the blackboard. Eventually, these three rendered images show occlusion correctly corresponding to their viewing positions, which are not on the image capturing line.

Next we present results for an outdoor scene, located in the landscape of the town YOKOHAMA. This experiment uses cylindrical images which are captured by some cameras positioned cylindrically as shown in Fig.1$-(b)$ on the car. The car runs on a public road in YOKOHAMA, capturing images and recording positions of those images by GPS. Then, we project them to cylindrical coordinates and store cylindrical images of each capturing point.

Our method rendered novel images viewing from a virtual running line, shown in Fig.8. Notice that these images are not viewed from the image capturing point, that is, these sequence of rendered images are viewed from another driving line on the road.

5 Conclusion and Future Work

This paper describes a new method for creating a 3-D large-scale virtual environment, for example, cities or towns. Mosaicing panoramic images captured by an omnidirectional camera or a measuring device of a similar type, our rendering system can create any view from an arbitrary point to an arbitrary direction on the ground in a virtual environment.

First, we capture panoramic images running along a straight line, and index the capturing position of each image. The rendering process is very simple and easy to compute. We need only select some suitable slits from stored panoramic images, and reassemble them to generate an image from a novel observation point. In other words, once images are recorded along a straight path, an arbitrary view around the path can be constructed.

Compared with similar methods of IBR, such as other plenoptic functions, this system has a great advantages. The main contribution of our method is that we can generate any view to any direction at any position on the ground; the disadvantage of similar approaches is that the range of field of view is relatively small. Thus, because it produces a large scale space, such as a virtual city, our system is outstanding. This method can also correctly render occluded objects, and where they are located. We plan to develop a driving simulator of the entire city of Tokyo using this method for the ITS purpose.

References

[1] S. B. Kang and R. Szeliski, 3-D scene data recovery using omnidirectional multibaseline stereo, in *Proc. of Computer Vision and Pattern Recognition Conf.*, pages 364-370, 1996

[2] S. Peleg and M. Ben-Ezra, Stereo panorama with a single camera, in *Proc. of Computer Vision and Pattern Recognition Conf.*, pages 395-401, 1999

[3] Y. Yu, P. Debevec, J. Malik and T. Hawkins, Inverse Global Illumination Recovering Reflectance Models of Real Scenes from Photographs, in *Proc. of ACM SIGGRAPH 99*, pages 215-224, 1999

[4] P. E. Debevec, C. J. Taylar, and J. Malik, Modeling and Rendering Architecture from Photographs: A hybrid geometry-and image-based aproach, in *Proc. of ACM SIGGRAPH 96*, pages 189-198. 1996

[5] S. K. Nayar, Omnidirectional Video Camera, in *Proc. of Computer Vision and Pattern Recognition '97*, page 482 - 488, 1997

[6] A. Lippman, Movie-Maps. An Application of the Optical Videodisc to Computer Graphics, in *Proc. of ACM SIGGRAPH '80*, pages 32-43, 1990

[7] S. E. Chen, QuickTime VR - An image-based approach to virtual enviroment navigation, in *Proc. of ACM SIGGRAPH'95*,pages 29-38, 1995

[8] E. H. Adelson and J. Bergen, The Plenoptic function and the elements of early vision, in *Computional Models of Visual Processing*,pages 3-20,MIT Press,Cambridge,MA,1991

[9] L. McMillan and G. Bishop, Plenoptic modeling: An image-based rendering system, in *Proc. of ACM SIGGRAPH'95*,pages 39-46, 1995

[10] M. Levoy and P. Hanrahan, Light field rendering, in *Proc. of ACM SIGGRAPH '96*,pages 31-42, 1996

[11] S. J. Gorther, R. Grzeszczuk, R. Szeliski, and M. F. Cohon, The lumigraph, in *Proc. of ACM SIGGRAPH*,pages 43-54, 1996

[12] T. Naemura, H. Harashima et al., Ray-Based Creation of Photo-Realistic Virtual World, *VSMM'97*, pages 59-68, 1997

[13] Heung-Yeung Shum and Li-Wei-He, Rendering with Concentric Mosaics, in *Proc. of ACM SIGGRAPH*, pages 299-306, 1999.

[14] M. Hirose, T. Endo, et al., Building a Virtual World from the Real World, in *Proc. of International Symposium on Mixed Reality.* (1999-3). 183-197

[15] H. Kawasaki, T. Yatabe, K. Ikeuchi and M. Sakauchi, Construction of a 3d city map using epi analysis and dp matching , Asian Conference on Computer Vision2000 (2000).

[16] Y. Onoe, K. Yamazawa, H. Takemura, and N. Yokoya, Telepresence by real-time view-dependent image generation from omnidirectional video streams, in *Computer Vision and Image Understanding, Vol.71, No.2*, pages 154-165, 1998.

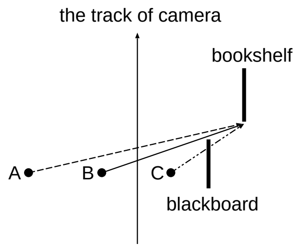


Figure 5: Rendered viewing positions (a),(b), and (c); synthesized images shown in Fig.8 are viewed from those points, respectively.

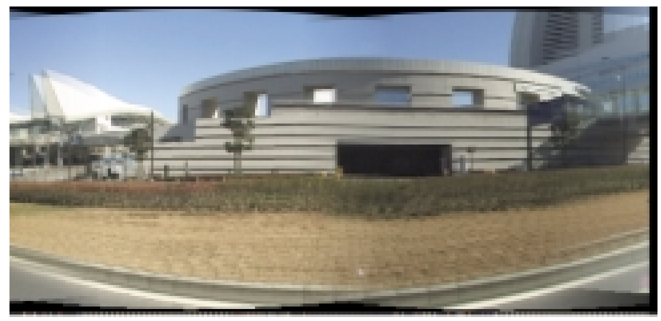
(a)

(b)

(c)

Figure 6: Rendered images; In (a), the left edge of the blackboard and the right edge of the bookshelf are separated. In (b), the left edge of the blackboard is lapped over the right edge of the bookshelf. in (c), the right edge of the bookshelf is occluded by the blackboard.

(a)



(b)

(c)

(d)

Figure 7: Rendered novel images viewed from virtual running line. These are views on another driving line.