Automatic Generation of Virtual Environment from Vertical Panoramic Image

Tze-kin LAO Kin-hong WONG Kam-sum LEE Siu-hang OR
{tklao,khwong,kslee,shor}@cse.cuhk.edu.hk
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China

Abstract

In this paper, a novel algorithm for creating virtual indoor environments is described. First, a panoramic mosaic is generated from a series of photos taken with a camera rotates along a horizontal axis. Then, from the panoramic mosaic image, a non-fixed viewing point virtual walkthrough system can be created by defining manually the corners in the vertical panoramic mosaic. The side ratio of the virtual walkthrough system can be obtained from the panning angle subsequently. By applying the cylindrical projection technique, the texture for the sides of the virtual walkthrough environment can be projected in a more realistic way. Real images have been used to verify our proposed algorithm with satisfactory results.

Keywords: Virtual Walkthrough Environment, Panoramic Mosaic, Cylindrical Projection

1. Introduction

In recent years, image-based rendering is commonly used by most computer graphic applications to construct virtual walkthrough environments. A collection of images is used to synthesize the scenes instead of building a complete 3D model of the objects. There are commercial software packages such as $Cool3D^{TM}$ and $PhotoVista^{TM}$ that provide such tools for users to create and navigate inside image-based panoramic virtual walkthrough environments. Chen [1] suggested three basic elements are the foundation of building a virtual walkthrough environment. First, a number of hot-spots of special interest are selected in the real environment. In each spot, a series of images is taken with respect to an observer standing at that spot. Second, these individual images are stitched together by some stitching algorithms [7] [8] [9] to form panoramas, and link up pairs of corresponding features among these hot-spot panoramas. Finally, images are texture mapped to become more realistic. A user can browse through links using panoramic viewers. Virtual navigation can be made possible by traveling from one node to another by clicking visual prompts (eg. icons) or messages showing that it is possible to jump to somewhere else.

Our proposed approach is defined here: Assuming we are viewing inside a long rectangle corridor and given several vertical panoramic nodes P_i with centers O_i representing the ith panoramic node and its center, r is the radius of the panorama, $\{\theta_{i1}, \theta_{i2}, \theta_{i3}, \theta_{i4}\}$ are the angles from the center to the corners of a real indoor environment of the ith node, estimate the ratio $\frac{H}{W}$ which describes the ratio between the sides of the real environment, and project the panoramas to the faces of the virtual walkthrough environment for browsing.

Instead of describing the whole process in details, this paper would concentrate on our approach combining with some existing modelling techniques in creating a virtual room from panoramas taken from real scenes so that users can walkthrough it as if in a real indoor environment. Figure 2 shows an overview of our entire system.

The next section describes our approach. Experimental results on real images are presented in the last section.

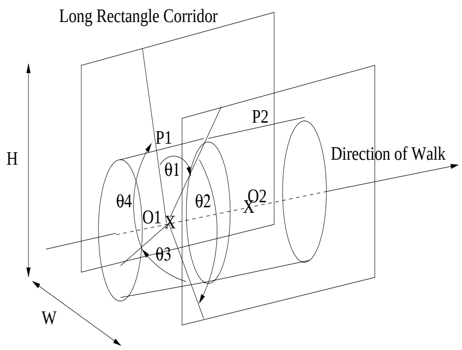

Figure 1. Definition on Our Approach

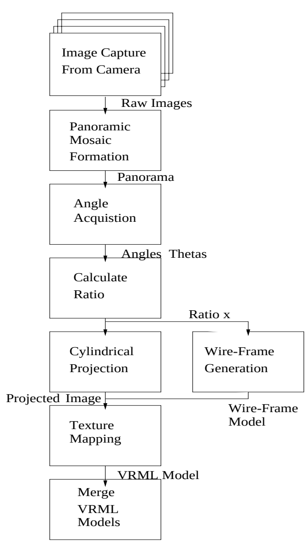

Figure 2. System Flow of our Approach

2. Image Manipulation

A series of images are taken from real environment using digital camera. They are taken along the horizontal axis. The position of camera is arbitary as shown in Figure 5.

2.1. Panoramic Mosaic Creation

After the image photos are obtained, panoramic mosaic is created based on existing mosaic creation algorithm. Many researchers have developed a number of techniques and software system for capturing panoramic images of real-world scenes. In particular, Chen [1] has developed a method requiring less hardware and regular photographic frames over the whole viewing space. Szeliski [9] pointed out that mapping 3D world coordinates (x, y, z) onto 2D panoramic coordinates (θ, v) with cylindrical projection is the first step to build a full view panorama:

$$\theta = tan^{-1}(\frac{x}{z}), v = \frac{y}{\sqrt{x^2 + z^2}} \quad (1)$$

where θ is the panning angle and v is the scanline. All the frames in a scene sequence are wrapped to construct mosaic images. Frame alignment with minor compensations for vertical jitter and optical twist is also needed for the construction process. The effect of camera motion have been cancelled out by various 2D or 3D parametric motion transformations [10]. These transformations also combined frames into complete panoramic images. In our current implementation, $Cool3D^{TM}$ was used to generate cylindrical mosaic system from 2D environmental snapshots. Information about horizontal translation t_x and vertical translation t_y for each input image are needed to estimate the incremental translational $\delta\mathbf{t} = (\delta t_x, \delta t_y)$ through a stitching algorithm which minimize the intensity error $E(\delta t)$ between two images. Figure 3a shows the mosaic segments constructed in our experiment.

2.2. Cylindrical Projection

After completed the construction of virtual mosaic images, image are wrapped onto a spheres or cylinder surface by using image viewer software utilizing texture-mapping. Only a portion of the panoramic

(a) (b)

Figure 3. (a)Vertical Mosaic Section (b)Cylindrical Projection Sections

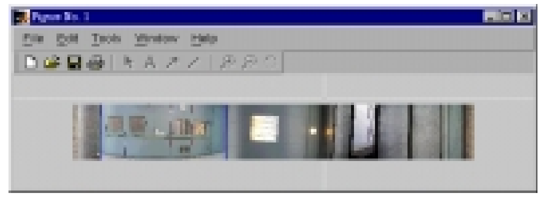

Figure 4. Corners Selection

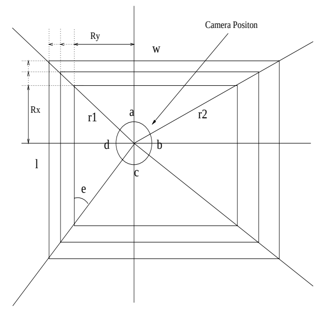

Figure 5. Cylindrical Projection Sections

image can be seen every time by a user on the image plane. A texture window is therefore formed by the bounding rectangle. The exact coordinates of the current windows can be found by projecting several points in the image plane onto the cylindrical surface and bounding the projected shape with a rectangle with the information of current viewing parameters under full-perspective projection model. Thus, the curved vertical edges in the panorama segment are rectified through this process.

2.3. Measurement of Panning Angles

Imagine what a user can observe when he/she is standing at the center of a panorama. As the environment is an indoor environment, the observer should have an angle when seeing the corners of the indoor environment. We need to have the angles before we build the virtual walkthrough environment for the indoor real environment. This can be done by two approaches. We can use a calibrated method by install a plate containing the reference angle so that we can measure the angle while taking photos. Also, we can measure the pixel number after generating the cylindrical wrapped image so that we can estimate the angle from the image. The latter one is used because the accuracy is higher and it is more convenience to do so.

When the panorama is generated from some commercially available software, i.e. $Cool3D^TM$, it can be fed into a matlab program to define the panning angle. Users can define the panning angle by selecting the corners in the indoor panorama. Figure 4 shows the selection of indoor corners. After choosing the corners, program uses the equation (2.1) to find the panning angles.

$$P.A.(i) = \frac{C(i+1) - C(i)}{T.W.} * 2\pi \quad for\ i = 1,2,3 \qquad (2.1)$$

$$P.A.(4) = \frac{(T.W. - C(4)) + C(1)}{T.W.} * 2\pi \qquad (2.2)$$

where P.A. is panning angle, C is corner selected, T.W. is the texture width. For the last panning angle, equation (2.2) is used. Thus, four panning angles are available for the estimation of side ratio.

2.4. Estimate Side Ratio

After obtaining the angles, we can work out equations so that the side ratio can be found. Given angles $\{a, b, c, d, e\}$ as shown in the Figure 5, side x and view distance $\{r_1, r_2\}$. By the Sine formula, we can define

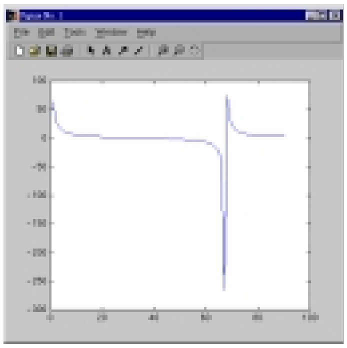

Figure 6. Plot of Ratio x w.r.t. angle e

three equations:

$$\frac{x}{\sin d} = \frac{r}{\sin e} \tag{3}$$

$$\frac{r_1}{\sin(3\pi/2 - a - d - e)} = \frac{1}{\sin a} \tag{4}$$

$$\frac{r_2}{\sin(c - e)} = \frac{x}{\sin b} \tag{5}$$

We can eliminate r_2 by

$$\frac{r_2}{\sin(-\pi/2 + d + e)} = \frac{1}{\sin a}$$

$$r_2 = \frac{\sin(-\pi/2 + d + e)}{\sin a} \tag{6}$$

Therefore, equation (5) becomes

$$\frac{\sin(-\pi/2 + d + e)}{\sin a * \sin(c - e)} = \frac{x}{\sin b} \tag{7}$$

As angles $\{a, b, c, d\}$ are known, and only three unknowns $\{x, r_1, e\}$ with three equation. We can solve the equations by iteration using some initial values. After we find the ratio x, we can generate a virtual environment with the side ratio x. Figure 6 shows the graph obtained from different value of ratio x with respect to angle e.

As we knew that the angle e should lie between 0 to $\pi/2$. We divide the range into 18 initial values and find the ratio x. There are a total of 18 possible solutions. Those unreasonable solution (eg. negative or very small) will be rejected. Table 1 shows all valid solution for 5 trials. They are also very close together as the root mean square error is small.

Trial	Ratio x	Angle e
1	1.38	0.65
2	1.40	0.65
3	1.34	0.66
4	1.39	0.66
5	1.39	0.66
Root Mean Sq. Error	1.52%	0.64%

Table 1. Size Ratio and Angle e

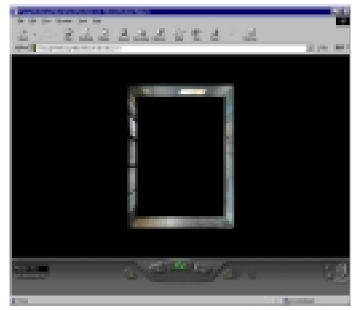

Figure 7. VRML result of a segment of corridor

3. Experimental Results

3.1. Virtual Environment Creation

For our virtual walkthrough environment creation system, with our Matlab 5.3 implementation executed on a Ultra- Sparc One, it takes about 20 minutes to synthesize a 9300x1000 pixels panorama. Although there are delays in generating an environment on-the-fly, the smoothness of walking through the environment is much superior than the algorithm aforesaid. Figure 7 shows the implementation result of creating virtual walkthrough environment. We can also merge several panoramas to form a larger environment as shown in the figure 8.

3.2. Horizontal rectangular environment

We have also implemented our approach on a horizontal rectangular environment. In fact most indoor environment floor plans, i.e., room, laboratory and office, are all rectangular in shape. Thus we can apply our system to them to create the virtual

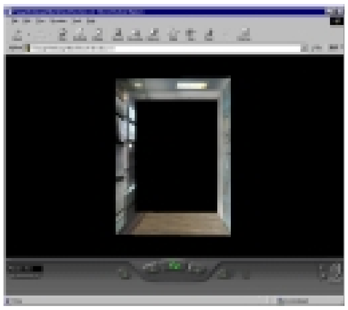

Figure 8. Merged VRML result of two segments of corridor

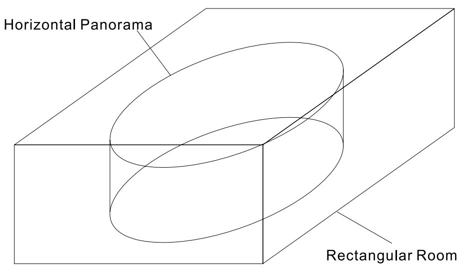

Figure 9. Horizontal rectangular environment

walkthrough environments. Since our system can estimate both the side ratio and camera position, we don't have to measure these dimensions by hand now. Figure 9 shows the idea of this application.

The calculation is just the same as we have described in previous chapters. The only thing different is that now the panorama is horizontal. Figure 10 shows different views of the resulting VRML model of a rectangular room. Table 2 shows the comparison between the actual side ratio and the estimated result. Moreover, table 3 shows the comparison between the actual camera position and the estimated result. The camera position coordinates are represented as a portion of the total length of each side as shown in figure 11. In both tables, we can observe that our estimations are close to the real data.

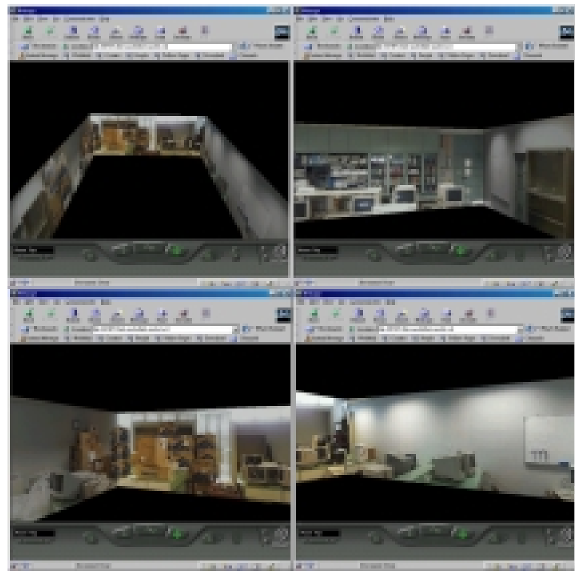

Figure 10. Resulting VRML model of the room

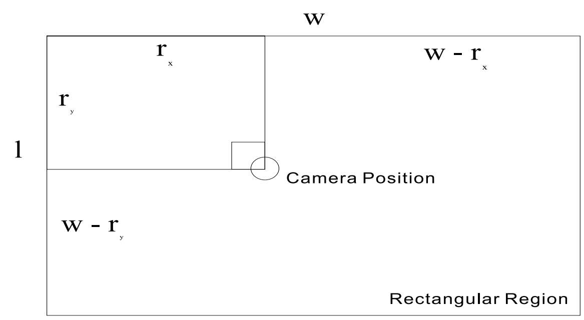

Figure 11. Camera Position

3.3. Building virtual environments for complex scenes

We may have some complicated environments in real scenes, which are not simply rectangular in shape. Thus we cannot apply our method directly. In such cases, we can usually decompose the complicated scene into several rectangles of different sizes and orientations. After the decomposition, we can simply apply our method onto each of those rectangles and create the models for each of them. Finally, we combine the models together according to the way they are separated and thus obtained the required virtual environment of the entire complicated scene. Figure 12 shows

Table 2. Comparison of Side ratios

	Side ratio $(l : w)$
Actual	1:1.786
Estimated	1:1.767

Table 3. Comparison of Camera positions

	Camera position $(r_x/w, r_y/l)$
Actual	$(0.422, 0.515)$
Estimated	$(0.420, 0.523)$

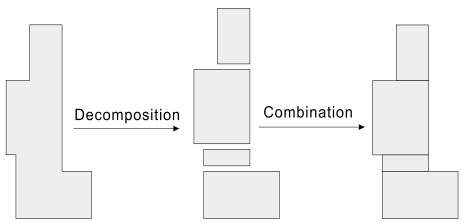

Figure 12. Creating complicated virtual environment by decomposition

the idea of creating complicated virtual environment by decomposition.

4. Discussion and Future Directions

We have implemented an algorithm to create a virtual walkthrough environment of an indoor scene. Though there is still room for improvements, our method can provide a realistic environment on the web for ordinary users with minimal sacrifice in image quality and execution speed.

The following are the significant advantages of our algorithm:

1. This algorithm requires very few human interaction and highly automated by computer. The users are only required to click on four points for the corners of the indoor environment, such that our program can proceed on the panning angle calculation, fine-tuning, side ratios estimation and virtual walkthrough environment generation. Some modelling tools require users to select contain hundreds pairs of corresponding points which are very inconvenient.

2. The computation time of this algorithm is very short compared with the existing 3D rendering tools under the same system configuration. Although the image quality would be a problem for faster implementation, this algorithm can be done on real-time after further enhancement. Moreover, since the environment is generated in VRML format, it can be put on the web for many commercial activities. (e.g. online shopping mall, virtual museum)

3. This algorithm provides a way to integrate the existing algorithms. Implementation of existing algorithm is an important method to evaluate the correctness of algorithm proposed. Also, this can enhance the flexibility of our proposed algorithm so that the algorithm can be used in various areas by plugging in different modules of different existing methods.

We are currently investigating the following possible enhancements:

1. There are two main features that our current implementation is relying on. They are noisy point correspondence and 3D feature dimensions. These measurements may not be readily accessible under all circumstances. Also, wrong calculation and unrealistic view synthesis may exist due to some noisy data. Our next target is to relax the dependency on point correspondences and feature dimension, or devise an update rule so that this information can be updated adaptively.

2. The 2D images currently used for constructing panoramic mosaic should have known focal lengths. This may led to different results for different focal lengths. Also, different size of each mosaic images (eg. one generate by images with resolution 640×480 and one generate by 1280×1024) may lead to difficulty on merging the virtual walkthrough environments generated. Possible solutions are using the same focal length and image size, or perform some alternation before using them in the construction process.

3. Other computer vision tools may be applied to enhance both speed and output quality in creating virtual environment. For instances, epipolar geometry [11] may be used to estimate the fundamental matrix **F** and transformation the 2D feature matching search space down to 1D. This can be done because there are some viewpoint differences among starting and ending nodes. Information about **F** with model fitting method like RANSAC [2] can help to drop out significant portions of false matches which effectively increase the image quality in synthesised views.

5. Conclusion

In this paper, a novel algorithm for creating virtual indoor environment is described. A panoramic mosaic is first generated with a series of images taken by an uncalibrated camera at an unknown position. Then a virtual walkthrough system can be created by first defining the corners in the panoramic mosaic. The side ratio of the virtual environment can be obtained from the panning angle, which is selected from the panoramic mosaic. We can also obtain the position of the camera. The texture for the sides of the virtual walkthrough environment can be projected to make it more realistic. Experiments on real images have been done to verify our approach. Further research work will focus on the improvement of image quality and speeding up of the program execution such that it can be used on real time application.

Acknowledgement

This research was supported by the Research Grant council of Hong Kong, under the UGC earmarked grant of Number CUHK4389/99E.

References

[1] Chen S.E. *QuickTime VR - An Image-based Approach to Virtual Environment Navigation,* Proc. of SIGGRAPH'95, pp.29-38.

[2] Fischler M.A. and Bolles R.C., *Random Sample Consensus: A Paradigm for Model Fitting with Applications to image Analysis and Automated Cartography,* Communications of the ACM, vol. 24, no. 6, pp. 381-395, June 1981.

[3] Lao T.K., Wong K.H., Lee K.S., Or S.H., *Automatic 3D-Object Modeling From Multiple Uncalibrated Images Using Active Contour* International Conference on Imaging Science, Systems and Technology, pp.176-182, 1999.

[4] Lippman A., *Movie Maps: An Application of the Optical VideoDisc to Computer Graphics,* Proc. of SIGGRAPH '80, pp. 32-43.

[5] Laveau S. and Faugeras O., *3D Scene Representation as a Collection of Images and Fundamental Matrices,* INRIA Technical Report No. 2205, Feb. 1994.

[6] Ripley D.G., *DVI - A Digital Multimedia Technology*, Communications of the ACM, 32(7), pp. 811-822, 1989.

[7] Shum H.Y. and Szeliski R., *Panoramic Image Mosaics,* Technical Report MSR-TR-97-23, Microsoft Research, 1997.

[8] Szeliski R., *Image Mosaicing for Tele-reality Application,* DEC Combridge Research Lab Technical Report, CRL94/2, May 1994.

[9] Szeliski R., *Video Mosaics for Virtual Environments,* IEEE Computer Graphics and Applications, pp. 22-30, Mar 1996.

[10] Szeliski R. and H.Y. Shum, *Creating Full View Panoramic Images Mosaic and Environment Maps,* SIGGRAPH'97, pp.251-258, Aug 1997.

[11] Xu G. and Zhang Z., *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach,* Kluwer Academic Publishers, 1996.