3D World from 2D Photos

Takashi Aoki, Tomohiro Tanikawa, and Michitaka Hirose

The University of Tokyo
7-3-1 Hongo, Bunkyo-ku,
Tokyo, Japan
{takashi,tani,hirose}@cyber.t.u-tokyo.ac.jp

Abstract. A large number of the world's cultural heritage sites and landscapes have been lost over time due to the progress of urbanization. Digital archive projects that digitize these landscapes as virtual 3D worlds have become more popular. Although a large numbers of studies have been made on reconstructing 3D virtual worlds, the previous methods have been insufficient, because they require significant effort. In this study, we propose a new method of reconstructing a 3D virtual world only from photo images that requires little intervention. The idea is to reduce the learning curve of the software need and automate the method as much as possible so that we can digitize as many heritage sites as needed. In our approach, we first reconstruct 3D models from single 2D photos using an image based modeling and rendering(IBMR) technique. After reconstructing models from all the available photos, we connect the 3D models into one unified 3D virtual world. Specifically, we implemented a seamless connection algorithm that supports free viewpoint translation. And we demonstrated the reconstruction of part of a cultural heritage site based on our system.

Keywords: digital archive, image based modeling and rendering, occlusion interpolation, 3D model seamless connection.

1 Introduction

Time and the progress of urbanization are destroying a large number of cultural heritage sites and landscapes. In response, many researchers or enterprises have begun to archive them as digital data. These digital archive projects have become increasingly popular. These landscapes are digitized as computer graphics. In addition, today's virtual reality technology has made it possible to have a highly realistic experience of immersion in such landscapes using the archived data. Such systems are very useful for not only research or entertainment, but also educational purposes such as learning history. For example, Ando et. al. created a history learning system for elementary school students and demonstrated its effectiveness [1].

However, we are currently not able to create such highly realistic computer graphics without significant effort and it needs many highly skilled computer graphic designers and researchers. In particular, computer graphic designers have to master various 3D graphic tools and such training incurs significant costs. This is the reason why digital archiving could be expensive. Thus, these methods have only been adopted only for profitable business purposes such as video games or cinema films.

M.J. Smith, G. Salvendy (Eds.): Human Interface, Part I, HCII 2007, LNCS 4557, pp. 249–257, 2007.

Furthermore, the IBMR technique has recently become an important area of research in computer graphic. IBMR has enabled the representation of a photorealistic view without the use of a highly complicated geometric 3D model. The method simply requires a large number of photo images and a few 3D geometry data.

There are many kinds of IBMR technique today. However an ideal method has not been proposed yet. Current methods have both advantages and disadvantages. A simple IBMR technique is image morphing. Seitz et. al. presented view morphing, which is the most well-known image morphing method [2]. And another the well-known IBMR technique is the light-field rendering[3][4]. This technique is based on the idea that we are able to represent an arbitrary view if the recording of all light rays passing through all positions and undergoing all rotations is possible. In this approach, a free viewpoint image is computed by interpolation among a large number of viewpoint images that are taken from a camera array system. Although these methods require only images, they have several weaknesses, for example, they require large-scale equipment such as camera arrays. There is another IBMR technique that constructs a 3D geometric model from a video. This concept is called structure from motion(SfM). An example is the factorization method proposed by Tomasi ans Kanade[5]. This method enables computation of image feature points, the follow up of in every frame and the computation of the 3D positions of the image feature points.

On the other hand, Hoiem et. al proposed the "Automatic photo pop-up" method [6]. This is the starting point of our work. Their algorithm enables to automatically digitize a 3D model from a single 2D photo image. This is a revolutionary method because although a 2D photo image has no geometric information, the algorithm can infer 3D data of the image. Today's advanced computer vision techniques and pattern recognition algorithms made it possible to turn a 2D image into a 3D environment. However, the quality of these 3D models depends largely on the learning result of object recognition. As a result, this method has not been used universally.

In this paper, we present the reconstruction of a 3D model from a single 2D photo using the IBMR technique. After reconstructing the models from all the photos taken, we connect these 3D models into a unified virtual environment. Specifically, we applied the seamless-connection algorithm to support free viewpoint translation. To demonstrate the effectiveness of our method to create a landscape digital archive, we actually digitize the Japanese transportation museum which was closed on 15 May 2006. This is a famous Japanese historical building, however, because of the deterioration of the building, the museum is being transferred to a new building. We have taken about 40000 pictures to archive the interior and exterior of the building. In this paper, we demonstrate the reconstruction of a small part of the building based on the image data.

2 Constructing 3D World

We present the method of reconstructing a 3D model from 2D photos in this section. We first describe the assumptions of the virtual 3D world. Next, we present the details of our 3D reconstruction algorithm.

2.1 Assumptions

In our approach, we set up the 3D model coordinates as follows.

- The positive Y-axis is the up direction and is perpendicular to the ground.
- The Z-axis is mapped on the shooting aspect line and the negative direction is toward the front.
- The X-axis is from left to right.

We assume that 3D models follow the following:

- The 3D model is constructed with a single horizontal ground surface.
- Foreground objects stand on the ground surface perpendicularly.
- The ground surface is always the plane. (y=0)
- A foreground object may be constructed from one or more vertical surfaces.
- The Angles of view (horizontal and vertical) are known.
- The height of the shooting viewpoint is known.
- The vanishing line in the photo image is known.

The first two assumptions have an important role in our approach, because they limit the variability of the calculated depth to only one value. Using this technique, we can specify the depth of foreground object surfaces. Figure 1 shows the concept of the 3D model reconstructed by our method.

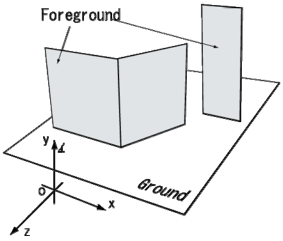

Fig. 1. Assumptions of 3D model

2.2 3D Reconstruction

We describe the method of 3D reconstruction from single 2D photos in this section. This approach requires two input images. One is a photo image and another is a segmented-region image which is created manually. Figure 2 shows an example of both images. From these images and parameters (angle of view, eye point height and vanishing line position), we compute the 3D geometry.
In particular, the segmented-region image is based on the following rules.

- The red region is the ground area in a photo image.
- The blue region is the sky area in a photo image.
- Other colors represent foreground objects in a photo image.
- A single surface in the 3D world is segmented with a single color.
- These colors do not represent metainformation such as the depth of a surface or other properties. They only represent an area in a image.

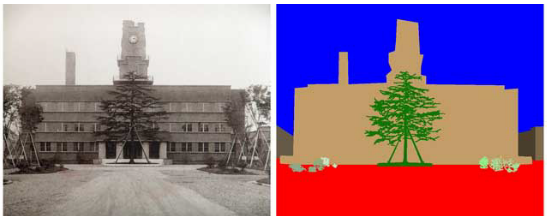

Fig. 2. The example of input images (left: photo image, right: segmented-region image)

Automatic image segmentation is currently a popular topic in computer vision, and much effort has been made to develop automatic image segmentation. However, images cannot be segmented with sufficient quality. Although the “Automatic Photo Pop-up” method has a full automated segmentation algorithm [6], it is not suitable for some situations. Thus, our approach requires little interaction during image segmentation.

To compute the 3D surface geometry, we first calculate the “feature edges” of each segmented region (shown in the left of Figure 3). The left and right edges are always vertical and located on the extreme left and right of the region respectively. The gradients of the top and bottom edges are computed by histogram analysis. As the right of shown in Figure 3, the sample data of this histogram are the gradients of line segments that are computed by subdividing the top or bottom curve. If the variance of the histogram is relatively small, the mode value is adopted for the edge gradient, otherwise the mean value is adopted. The main reason for this is to compute the precise feature edges of a irregular object such as a tree or a person.

Feature edges are important role for computing 3D geometry. In particular, the bottom edge represents a tangential line with the ground and is used to compute the depth of a surface.

Next, we compute the depth of a foreground object surface from the feature edges. In Figure 4, we show the 3D geometric relationship between a 2D photo and a 3D foreground object.

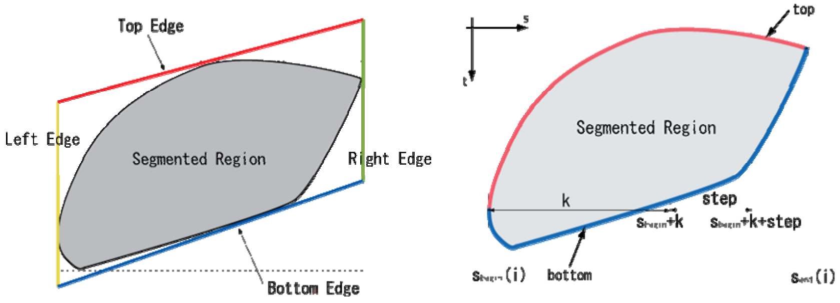

Fig. 3. Feature edges (left: concept, right: how to compute edge gradient)

As shown in Figure 4, we can calculate the depth of a foreground object surface Z_{3d} :

$$Z_{3d} = L \tan \psi \tag{1}$$

where L is the eye point height. Also θ , ϕ and ψ can be computed:

$$\theta = \arctan \frac{h}{(H/2)(1/\tan(\alpha_{Vaov}))}, \quad \phi = \arctan \frac{h_g}{(H/2)(1/\tan(\alpha_{Vaov}))}, \quad \psi = \frac{\pi}{2} - (\theta - \phi) \tag{2}$$

Where H is the photo image height and α_{Vaov} is the vertical angle of view. In the same way, other vector elements can be computed.

Furthermore, we can calculate texture map coordinates by projecting the 2D photo image to the 3D geometry model. However, this will not be described in this paper due to lack of space.

Finally, we create texture images for each object (the ground, sky and foreground objects).

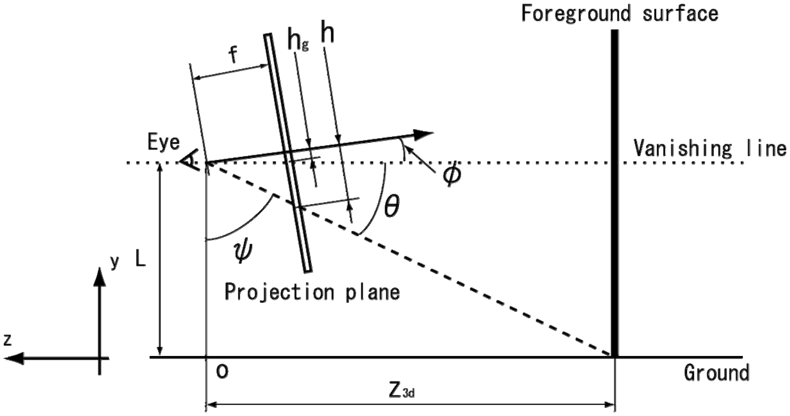

Fig. 4. Geometric relationship

2.3 Occluded-Area Interpolation

In our method, we apply texture image interpolation in accordance with the depth of the surface. A 3D model constructed by the method described in 2.2 has occluded texture image gaps. This approach fills these occluded gaps and makes it possible to support the free viewpoint translation and rotation in the virtual world.

There are two possible situations that we must consider before we fill the texture gap. One situation is when the texture image is occluded by a foreground object resulting in a texture gap. The other situation is when the texture image contains a gap or a hole. We employ an algorithm that can check whether the gap is the result of an occlusion. If it is the result of an occlusion, we employ the occluded-area interpolation algorithm to fill the gap.

This algorithm is simple. We first create a filling map that indicates the occluded area in a texture image. Figure 5 shows the concept of our algorithm. We scan a texture image over each coordinate (s and t). If there are colored pixels on a scan line and gaps between the extreme left and right, we indicate the gaps as interpolation candidates and we compute the depth value of each candidate pixel for regions i and j (see Figure 5). Comparing these two depth values, if the depth of region i is larger than that of region j, the candidate is occluded by an object in region j, otherwise it is not occluded and region i has a gap or a hole. By applying this algorithm to all the texture images, we can compute interpolation maps.

We fill the texture image gaps in accordance with the interpolation maps. We employ the back projection for lost pixels(BPLP) method on the eigenspace" as an interpolation algorithm [7]. This algorithm fills image gaps based on the local self-similarity of an image, i.e., one local region in an image is similar to another local region. The reason why we chose the BPLP method is that it only requires a source image as input information and takes relatively little time to carry out the process. In our approach, we must interpolate more than one texture image; thus, this algorithm suits our approach because the process is simple and relatively quick.

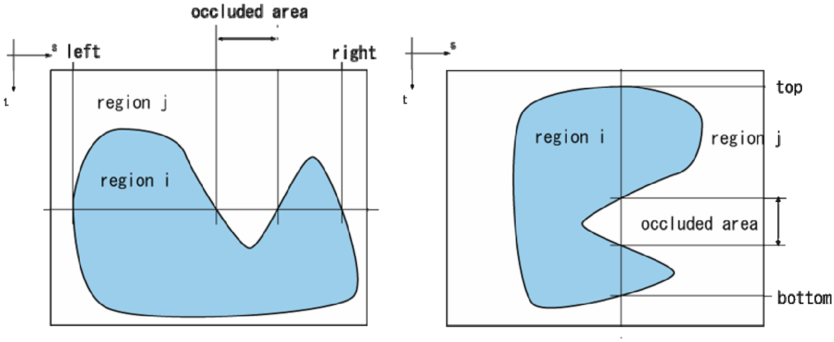

Fig. 5. Concept of interpolation map computation

2.4 Seamless Connection

A 3D model reconstructed from single 2D images has a small view-translation capacity. To solve this problem, we propose a 3D model connection method. Our approach is to switch seamlessly from one 3D model to another, similar to a picture-story show. By switching in accordance with the user's view-point position, we can represent a large 3D world.

Our method has 3 steps:

1. Link a 3D model and its metainformation (the position and rotation from where the photo is shot).
2. Search the 3D model from a database in accordance with the user's view position and rotation.
3. Seamless rendering.

We first link a 3D model and its metainformation (the shot position and rotation). This information is required for the searching at the next step. In our approach, we link the model by describing a 3D model data file path and its metainformation in one XML file.

Next, we search the displayed 3D models in accordance with the user's view information and 3D model metainformation. Our search method is based on an evaluation function. We calculate evaluation values for all 3D models and display a certain number of top scoring 3D models. The evaluation functions adopted here are

$$D_i = \cos\left(\frac{\sqrt{(x_e - x_{mi})^2 + (y_e - y_{mi})^2}}{P_{radius}} \cdot \frac{\pi}{2} \right) \tag{3}$$

$$R_i = \cos\left(\frac{1 - \cos(\theta_e - \theta_{mi})}{P_{angle}} \cdot \frac{\pi}{2} \right) \tag{4}$$

D_i evaluates the positional distance and R_i evaluates the rotational distance. (x_e, y_e) and θ_e are the user's view point position and rotation, respectively. And $(x_{mi}, y_{mi})$and θ_{mi} are the 3D model (indexed i) position and rotation, respectively. P_{radius} andP_{angle} are parameters. These parameters depend on the 3D model dominant density in the virtual 3D environment. If the density of an area is high, these parameters should be set small; otherwise they should be set large. The output evaluation value is the product of D_i and R_i .

For rendering, we use the alpha-blending technique for connection. In this method, the blending ratio is computed by normalizing the evaluation values of the displayed 3D models. In addition, we edit the texture image alpha values:

$$\alpha(x, y) = A_x \sin\left(\frac{x}{W/2} \cdot \frac{\pi}{2}\right) \cdot A_y \sin\left(\frac{y}{H/2} \cdot \frac{\pi}{2}\right) \tag{5}$$

where W is the image width, H is the image height andA_x and A_y are the peak values of alpha. This operation makes it possible to form seamless boundaries, because the sin function is twice differentiable and people cannot recognize the differences in an overlapping area.

3 Experiments and Results

3.1 Occluded-Area Interpolation

Figure 6 shows the results of 3D reconstruction (without and with interpolation) from a single 2D photo. This demonstrates that our occlusion interpolation method is effective.

In our implementation, which uses OpenCV and Lapack libraries, it takes about 5 minutes to reconstruct a 3D model by including interpolation starting from a 3264x2448 input image on a Pentium 4 3.2 GHz computer. However, without interpolation, it takes about 10 seconds or less. This is because the interpolation method requires a principal component analysis(PCA) of a large number of dimensions.

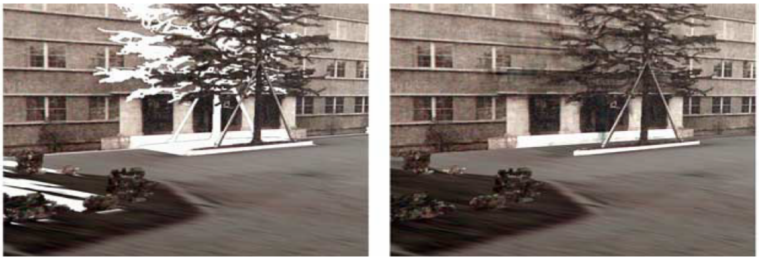

Fig. 6. Result of 3D reconstruction (left: without interpolation, right: with interpolation)

3.2 3D Virtual World

Figure 7 shows images of the 3D virtual world. This demonstrates the effectiveness of our method at reconstructing a photorealistic 3D virtual world.

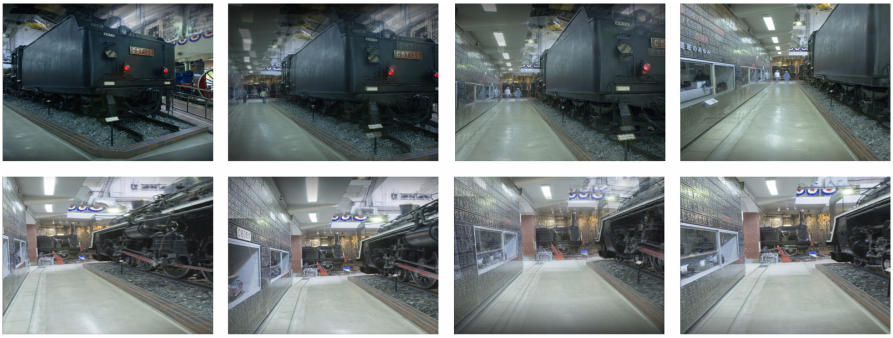

Fig. 7. Images of 3D virtual world showing seamless connection

4 Conclusion

In this paper, we proposed a method that can reconstruct a 3D virtual world only from photo images. By digitizing some parts of the Japanese transportation museum, we demonstrated the effectiveness of our method at digital archiving.

Our occluded-area interpolation algorithm is simple, but it can fill gaps in the texture image and makes it possible to remove the view translation and rotation limitations in 3D virtual worlds. Furthermore, our seamless connection method allows users to walk around large-space 3D virtual worlds.

One future aim of this research is to improve the 3D world quality. The 3D model reconstructed using our method cannot represent a curved surface and has a few computational errors under certain circumstances. In addition, there are some situations for which the BPLP interpolation method does not work well.

Another direction for future research will be its expansion into the use of the World Wide Web. This would allow the development of community spaces dedicated to photorealistic 3D virtual space creation to encourage the participation of the general public.

References

1. Ando, T., Yoshida, K., Tanikawa, T., Wang, Y., Yamashita, J., Kuzuoka, H., Hirose, M.: Proto-type Educational Contents by using Scalable VR System Historical Learning in Copan Ruins of Maya Civilization. In Trans. VRSJ, vol. 8(1) (2003)
2. Seitz, S.M., Dyer, C.R.: View Morphing. In: Proc. 23rd Annual Conf. Computer Graphics and Interactive Techniques, pp. 75–82 (1996)
3. Levoy, M., Hanrahan, P.: Light Field Rendering. In: Levoy, M., Hanrahan, P. (eds.) Proc. 23rd Annual Conf. Computer Graphics and Interactive Techniques, pp. 31–42 (1996)
4. Gortler, J,S., Grzeszczuk, R., Szeliski, R,, Cohen, F.M.: The Lumigraph. In: Proc. 23rd Annual Conf. Computer Graphics and Interactive Techniques, pp. 43–54 (1996)
5. Tomasi, C., Kanade, T.: Shape and Motion Without Depth. In: Proc. 3rd International Conf. Computer Vision, pp. 137–154 (1990)
6. Hoiem, D., Efros, A.A., Heber, M.: Automatic Photo Pop-up. In: Proc. ACM SIGGRAPH 2005, pp. 577–584 (2005)
7. Amano, T., Sato, Y.: Image Interpolation Using BPLP Method on the Eigenspace. in Trans. IEICE (D-II) J85(3), 457–465 (2002)