

# Augmenting panoramas with object movies by generating novel views with disparity-based view morphing

Yi-Ping Hung\*, Chu-Song Chen, Yu-Pao Tsai and Szu-Wei Lin

Our goal is to augment a panorama with object movies in a visually 3D-consistent way. Notice that a panorama is recorded as one single 2D image and an object movie (OM) is composed of a set of 2D images taken around a 3D object. The challenge is how to integrate the above two sources of 2D images in a 3D-consistent way so that the user can easily manipulate object movies in a panorama. To solve this problem, we adopt a purely image-based approach that does not have to reconstruct the geometric models of the 3D objects to be inserted in the panorama. A critical issue of this method is how to generate the novel views required for showing an OM in different places of a panorama, and we have proposed a view morphing technique, called *t*-DBVM, to solve this problem. Our experiments have shown that this purely image-based approach can effectively generate visually convincing OM-augmented panoramas. This method has great potential for many applications that require integration of panoramas and object movies, such as virtual malls, virtual museum, and interior design. Copyright © 2002 John Wiley & Sons, Ltd.

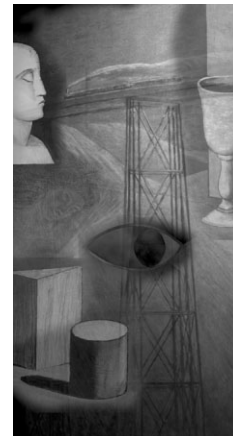
Received: ???; Revised: 2 May 2002

KEY WORDS: panorama; object movie; augmented reality; image-based rendering; novel view generation; disparity-based view morphing

## Introduction

This paper is concerned with the construction of a special class of augmented reality, which will be referred to as *augmented panoramas*. In augmented panoramas, some foreground objects are inserted into a background scene recorded in the form of panoramic images. This background scene can be either static or dynamic. If the background scene is static, we can first acquire a sequence of overlapping images and then produce a panoramic image by using image-mosaicking techniques.<sup>1,2</sup> If the background scene to be recorded is a dynamic one, we can capture it with omnidirectional cameras.<sup>3</sup> In fact, the panoramic background can also be a computer-generated scene. The advantage of using panorama to record a computed-generated virtual scene is that the rendering can be real time no matter how complex the virtual environment is.

\*Correspondence to: Y.-P. Hung, Institute of Information Science, Academia Sinica, Nankang 115, Taipei, Taiwan.  
E-mail: hung@iis.sinica.edu.tw



Augmented panoramas have many useful applications, such as virtual mall, virtual museum, and interior design. For example, for Internet shopping in a virtual mall, different commodities can be exhibited, in different seasons or for different categories of customers, on the same counter of an existing and well-decorated department store.

Inserting a '3D object' into a '3D scene' would be an easy task, provided that we have the geometric models of both the 3D object and the 3D scene. However, reconstruction of 3D geometric models is known to be a tedious and expensive job, and one may want to avoid it if possible. On the contrary, panoramas and object movies are relatively easy to create, and can yield photo-realistic results. Examples of panoramas and object movies can be found in <http://www.apple.com/quicktime> and many other websites. In this paper, our goal is to insert an object movie into a 2D panoramic image *in a visually 3D-consistent way*, i.e., the resulting visual effect looks geometrically consistent in the 3D space. Notice that an object movie (OM) is in fact a set of images taken from different perspectives around a 3D object. Hence,

the issue becomes a problem of integrating different sources of 2D images, in a visually 3D-consistent way, *without reconstructing their 3D geometric models*. This is not a trivial problem, if we do not have 3D information of the image-based objects and the panoramic scene.

A similar problem, which inserts 3D geometry-based objects into images or videos, is usually referred to as augmented reality and has received much attention in recent years.<sup>4-7</sup> To insert virtual objects into a 'real' video (i.e., a video capturing real-world or real events), a major difficulty is to estimate the parameters of the real camera. The parameters can be estimated by using reference points or patterns,<sup>4,5</sup> or without using any reference patterns.<sup>6,7</sup> Instead of inserting 3D objects into videos, as in most augmented reality applications, Chen *et al.*<sup>8</sup> inserted 3D objects into *one single image* by utilizing a user-specified cuboid structure. Notice that there is no inter-frame information available when the augmentation is done with one single image. Another related work is that by Katayama *et al.*,<sup>9,10</sup> which discussed how to manipulate multiple perspective images of an object within a 3D geometry-based scene using some prior knowledge such as feature points, base planes, or bounding volumes. However, to our best knowledge, no successful work has been reported previously on integrating image-based objects into an image-based panoramic scene.

This paper presents an easy-to-use approach to augmenting a panorama with OMs, in such a way that the inserted foreground objects, recorded as OMs, *always* remain visually 3D-consistent with the panoramic background—even when the users are interactively manipulating the OMs in front of the background. In this work, the places in the panorama where the OMs are to be inserted are specified by the user, or more precisely, by the author of the OM-augmented panorama. Consider the example shown in Figure 1. The two places selected for inserting OMs are a tabletop and a countertop. Figure 1(a) shows the entire panoramic image of the panorama to be augmented. Figures 1(b) and (c) show the two places selected for the insertion. Suppose our task is to insert a Winnie into Figure 1(b) and a Kitty into Figure 1(c). Then, the two snapshots shown in Figures 2(a) and (b) exemplify the initial status of the augmented views. Notice that the object to be inserted into the panorama is recorded in the form of an OM, which is actually an interactive movie. Hence, we should allow the user to interactively manipulate the object in front of the panoramic background. Figures 2(c) and (d) illustrate another two perspectives of the objects after the user rotates the object horizontally.

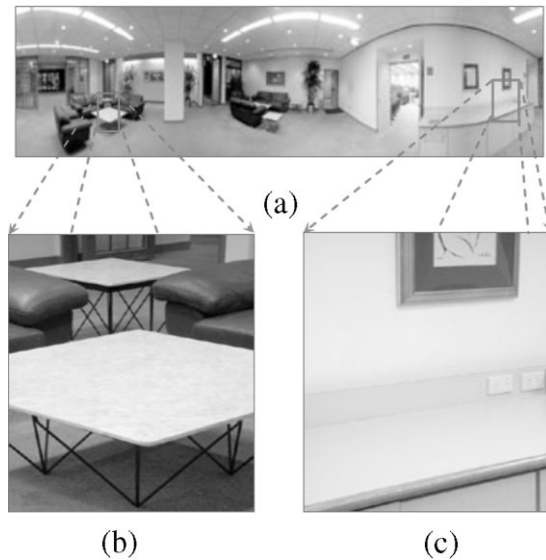


Figure 1. (a) Shows a complete panoramic image for augmentation; (b) and (c) exemplify the two places selected for inserting object movies.

The remainder of this paper is organized as follows. The next section describes how the images comprising the OMs are acquired, and then explains why we need to generate novel views and how we achieve it with the trinocular disparity-based view morphing (*t*-DBVM). The third section presents the image-based method

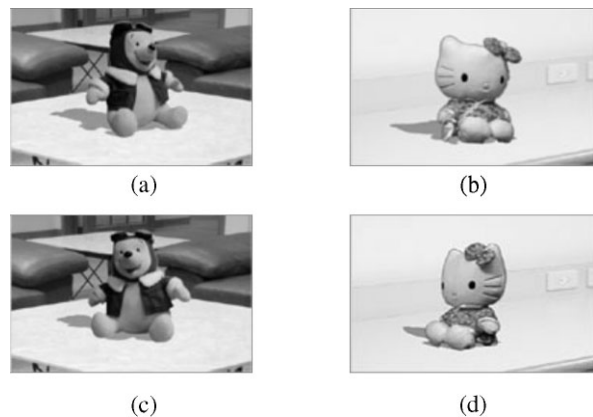


Figure 2. (a) and (b) Two snapshots of the augmented panorama, which corresponds to Figures 1(b) and (c), respectively; (c) and (d) show another two perspectives of the objects after the user rotates the object horizontally. Notice that our method for the automatic image composition is purely image-based in the sense that there is no need to reconstruct the geometric models of the 3D objects while we are still able to render the visually convincing shadows.

used to integrate OMs with a panorama in a visually 3D-consistent manner. The fourth section shows some of our experimental results. The final section gives a conclusion and some directions for future work.

## Object Movies

### Image Acquisition

In this paper, the set of images that comprise an OM are captured by using a digital camera mounted on Kaidan's Magellan M-2500, as shown in Figure 3. By using this image capture device, we know the pan and tilt angles,  $\theta$  and  $\phi$ , of the viewing direction associated with each captured image (though the angle values need not to be very precise with our method). In this set-up, the optical axis of the camera roughly points toward the *rotation centre* of Magellan M-2500, which is the intersection point of the pan axis (for rotating the object) and the tilt axis (for rotating the camera). When photographing an OM, this rotation centre will reside at a certain point with respect to the object being photographed. This point will then become the rotation centre of the object when a user is browsing the OM, and will be denoted by  $C_O$ .

In this paper, we define the object reference frame (ORF) to be the coordinate system locating at  $C_O$ , with its  $x$ -axis pointing to the camera while the camera is taking the first image (i.e.,  $\theta = 0$  and  $\phi = 0$ ) and its  $z$ -axis pointing upward.

A few samples of the captured images that comprise the OM of Winnie are shown in Figure 4. It can be seen from Figure 4 that the sampled images were captured at discrete viewpoints. If we increase the number of sampled images, the playback of the OM will become smoother but the amount of image data will also in-

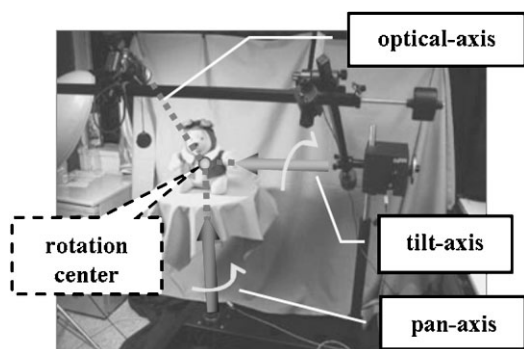


Figure 3. The experimental set-up for acquiring images of object movies.

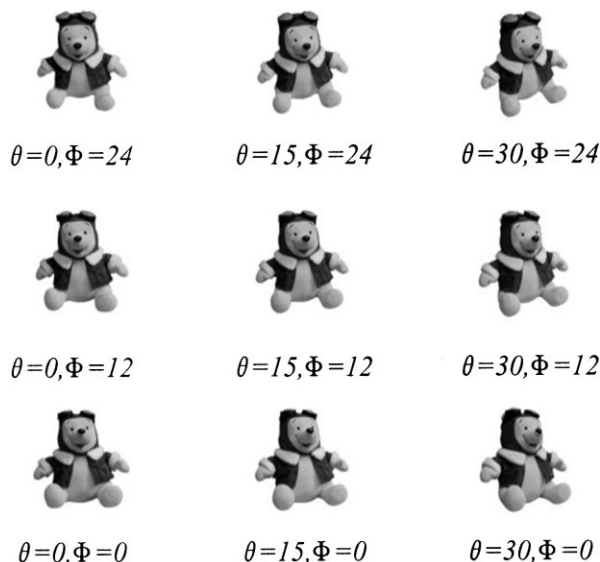


Figure 4. A few samples of the multi-perspective images that comprise an object movie of Winnie.

crease accordingly. One approach to solving this dilemma is to adopt the techniques for novel view generation (or view interpolation), which uses a finite set of sampled images to create an infinite number of intermediate views.

In augmented panoramas, novel view generation is desired not only because the playback of an OM can be smoother given the same amount of acquired images, but also because the intermediate views are indispensable, as explained below. Consider Figure 5, where our initial intention is to insert the object at point  $Q_1$ . In this case, the viewing direction should be  $\overrightarrow{C_P Q_1}$ , where  $C_P$  is the viewer centre of the panorama. Hence, the tilt angle for photographing the object should be  $\phi_1$ . Now, suppose the author of the augmented panorama would like to move the object closer to the viewer, e.g., moves the object to point  $Q_2$ . Then the system will have to create a

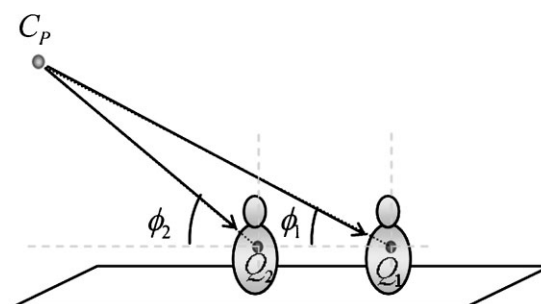


Figure 5. Moving the object around will change the viewing perspective of the object.

new image of the object that is supposed to be seen if this object were to be photographed from the tilt angle  $\phi_2$ . The same situation happens if the author would like to put the object on a higher or lower tabletop instead. Also, the pan angle will vary if he would like to move the object further left or further right. This is the reason why an OM to be integrated with a panorama should be a  $360^\circ \times 360^\circ$  object movie, or at least a  $360^\circ \times 180^\circ$  one. For an OM that was taken only with a fixed tilt angle (i.e., a single-axis-rotation OM), the visual effect of 3D geometric consistency can be hardly achieved in the above situation.

There are many methods available for novel view generation. Light field rendering<sup>11</sup> and lumigraphs<sup>12</sup> are two famous ones, but their large memory requirements make them impractical for real applications, especially for those requiring Internet transmission. Another approach tries to reconstruct some geometric model of the object based on consistency with the image information, such as Photo Hull.<sup>13</sup> This approach seems to be quite promising for reducing the memory size of an OM, but the camera parameters have to be known accurately for reconstructing the 3D shape. In this paper, we adopt disparity-based view morphing (DBVM) for generating the novel views, which does not require accurate camera parameters.

### Novel View Generation Using *t*-DBVM

View morphing<sup>14</sup> is an image-based rendering technique proposed for generating realistic intermediate views from two images for lateral camera motion (i.e., when the camera viewing direction is perpendicular to the camera motion). To avoid the limitation of lateral camera motion, Huang *et al.* have extended the view morphing technique to disparity-based view morphing<sup>15</sup> (abbreviated as DBVM), which allows general camera motion. In this paper, in order to generate novel views for an OM, we further extend the DBVM to the *t*-DBVM (an abbreviation of *trinocular* DBVM), which can synthesize intermediate views from *three* observed views.

Consider the image acquisition of an OM by using Magellan M-2500 shown in Figure 3. We can imagine that all of the positions of the camera are placed on the surface of a pseudo sphere, called the *image-acquisition sphere*. Due to the limitation of time and space, only a finite set of discrete viewpoints can be used for taking real images of the object. The intermediate views between those samples will then be generated by using the *t*-DBVM technique, as described below.

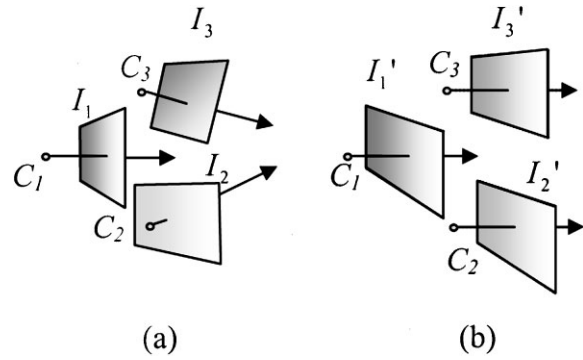


Figure 6. Illustration of image rectification. (a) The original views. (b) The rectified views.

Given a desired viewing direction, we first find out the three nearest views of this viewing direction. Notice that those three views are not supposed to lie on the same longitude or latitude of the pseudo sphere. Then, by applying the *t*-DBVM, the desired view can be interpolated from the three nearest views, as described in the following four steps.

**Step1: Rectification.** In this step, we rectify the three nearest views to be parallel with each other, as shown in Figure 6, where  $C_i, i = 1, 2, 3$ , are the camera centres of the three views with respect to the world coordinate system, and  $I_i$  and  $I'_i, i = 1, 2, 3$ , are the original images and the rectified images associated with the three views, respectively. Let  $\Pi_i$  denote the projection matrix associated with the *i*th view. Then, by using the pinhole camera model, we have

$$\begin{aligned}
 p_i &\sim \Pi_i P \\
 \Pi_i &= K[R_i t_i]; \quad i = 1, 2, 3 \\
 t_i &= -R_i C_i
 \end{aligned} \tag{1}$$

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where  $P$  is a 3D point,  $p_i$  is the projection of  $P$  on the image  $I_i$ ,  $K$  and  $[R_i t_i]$  represent the intrinsic and extrinsic parameters associated with the *i*th view, respectively,  $f$  is the focal length,  $(u_0, v_0)$  is the image centre (or piercing point), and  $'\sim'$  means equality up to scale, i.e.,  $\lambda \hat{p}_i = \Pi_i P$ , where  $\hat{p}_i = (p_i^T, 1)^T$ .

In our method, the viewing direction of the rectified views is set to be the one parallel to the normal of the triangle  $\triangle C_1 C_3 C_2$  and the *x*-axis is chosen to be the one

parallel to the vector  $\overrightarrow{C_1C_2}$ . Since the camera centres of the rectified views remain the same as those of the original views, the projection matrix associated with the rectified views can be described as follows:

$$\begin{aligned} p'_i &\sim \Pi'_i P \\ \Pi'_i &= K[R_r | t'_i]; \quad i = 1, 2, 3 \\ t'_i &= -R_r C_i \end{aligned} \quad (2)$$

where  $p'_i$  is the projection of a 3D point  $P$  on the rectified image  $I'_i$ , and  $R_r$  is the rotation matrix of the three rectified views. Then, we can derive the following equation from Equations (1) and (2) for transforming the original image to the rectified image:

$$\begin{aligned} p'_i &\sim H_i p_i; \quad i = 1, 2, 3 \\ H_i &= KR_r R_i^{-1} K^{-1} \end{aligned} \quad (3)$$

where  $p_i$  and  $p'_i$  are corresponding points, i.e., they are the projections of the same 3D point  $P$  in the original image  $I_i$  and in the rectified image  $I'_i$ , respectively. Notice that the above equation is a homography, because each rectified view has the same camera centre as its original view.

**Step 2: Disparity Estimation.** In  $t$ -DBVM, the intermediate views are rendered based on the disparity maps estimated from the corresponding images. In this step, we extend the method proposed by Huang *et al.*<sup>16</sup> for disparity estimation. The major improvement in this paper is that, in the disparity refinement step, we add an energy term for anchor points into the energy function for global optimization. The anchor points are the accurate correspondence points that can be automatically detected by the corner detection algorithm, or even manually selected by human interaction. Therefore, during disparity refinement, the anchor points are fixed in their location and can affect their neighborhood to get more robust disparity maps. At the end of this step, six disparity maps  $D_{i,j}$ ,  $i, j = 1, 2, 3$ , and  $i \neq j$ , can be obtained for real-time generation of novel views.

**Step 3: Linear View Interpolation.** Seitz and Dyer<sup>14</sup> have shown that, after a pre-warping (or rectification) step, the intermediate views can be obtained by linearly interpolating the positions and colors of corresponding image points. Here, we extend the linear interpolation from two views to three views. Let  $p'_1, p'_2$ , and  $p'_3$  be the projections of a 3D point  $P = (X, Y, Z, 1)^T = (V_P^T, 1)^T$  in the three rectified images,  $I'_1, I'_2$ , and  $I'_3$ , respectively.

Suppose the projection matrix of the desired view can be expressed as a linear interpolation of the projection matrices of  $I'_1, I'_2$ , and  $I'_3$ ,  $\Pi_n' = w_1 \Pi'_1 + w_2 \Pi'_2 + w_3 \Pi'_3$ , where  $0 \leq w_1 \leq 1, 0 \leq w_2 \leq 1, 0 \leq w_3 \leq 1, w_1 + w_2 + w_3 = 1$ . Then, the projection of  $P$  in the desired view can be expressed as a linear interpolation of  $p'_1, p'_2$ , and  $p'_3$ , as shown below:

$$\begin{aligned} p''_n &\sim \Pi'_n P \\ &= (w_1 \Pi'_1 + w_2 \Pi'_2 + w_3 \Pi'_3) P \\ &= K[R_r | -R_r(w_1 C_1 + w_2 C_2 + w_3 C_3)] P \\ &= K[R_r | w_1 t'_1 + w_2 t'_2 + w_3 t'_3] P \\ &= (w_1 + w_2 + w_3) KR_r P + K(w_1 t'_1 + w_2 t'_2 + w_3 t'_3) \\ &= w_1 KR_r P + w_1 K t'_1 + w_2 KR_r P + w_2 K t'_2 + \\ &\quad w_3 KR_r P + w_3 K t'_3 \\ &= w_1 K[R_r | t'_1] P + w_2 K[R_r | t'_2] P + w_3 K[R_r | t'_3] P \\ &= w_1 \Pi'_1 P + w_2 \Pi'_2 P + w_3 \Pi'_3 P \end{aligned}$$

Notice that the third components of  $\Pi'_i P$ ,  $i = 1, 2, 3$ , are all the same, because they are actually the depth of  $P$  after transformed to the coordinate systems of the rectified views. Therefore, we have

$$p''_n = w_1 p'_1 + w_2 p'_2 + w_3 p'_3 \quad (4)$$

One method to express the projection matrix of the desired view is to define a 2D reference frame by using the camera centres ( $C_1, C_2$  and  $C_3$ ) of the three observed views and let  $w_1, w_2$  and  $w_3$  be the barycentric coordinates of the desired view falling within the triangle  $\triangle C_1 C_3 C_2$ . Given the weighting factors which specify the desired view, we can then warp the three rectified images  $I'_i$  to  $I''_i$ ,  $i = 1, 2, 3$ , based on the disparity maps  $D_{i,j}$ ,  $i, j = 1, 2, 3$ , and  $i \neq j$ . Next, we blend the warped images,  $I''_1, I''_2$ , and  $I''_3$  to form the rectified image of the desired view, which considers single occlusion.<sup>15,16</sup> This blended image will be referred to as  $I_n'$ .

**Step 4: De-rectification.** After performing the linear view interpolation in step 3, what we obtain is the rectified image  $I_n'$  of the desired view, which can then be de-rectified as follows. First, we compute the rotation matrix  $R_n$  of the desired view by interpolating the rotation parameters of the three observed views based on the weighting factors,  $w_1, w_2$  and  $w_3$ . Then, the de-rectification can be achieved by using the inverse of Equation (3) while replacing  $R_i$  with  $R_n$ . It should be noticed that the camera centre of the interpolated view obtained thus far is not lying on the image-acquisition sphere containing  $C_1, C_2$ , and  $C_3$ , but rather lying on the

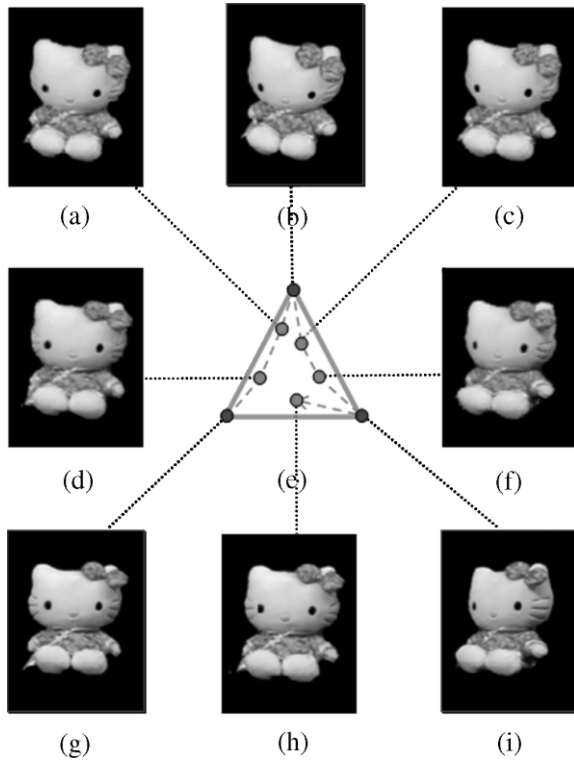


Figure 7. Illustration of novel view generation using *t*-DBVM: (b), (g) and (i) show the source views; (e) shows the reference frame used for indexing the desired views; (a), (c), (d), (f) and (h) are the novel views corresponding to the circles within the triangle shown in (e).

triangle  $\triangle C_1 C_3 C_2$ . Hence, the rendered image needs to be farther scaled down accordingly in order to achieve a smoother browsing effect.

Figure 7 shows a few examples of the novel views generated by using the *t*-DBVM described above. Figures 7(b), (g) and (i) are the three observed views used to generate the novel views. Figure 7(e) shows the reference frame for indexing the camera centre of the novel views. To browse an object, the user may drag the cursor to follow the dashed line shown in Figure 7(e). Then, Figures 7(a), (c), (d), (f) and (h) are the novel views that the *t*-DBVM will generate when the cursor is located at those circles shown in Figure 7(e).

## Integration of OMs with Panoramas

To insert an OM into a panorama, our approach requires the user to first select a 3D shadow reference frame

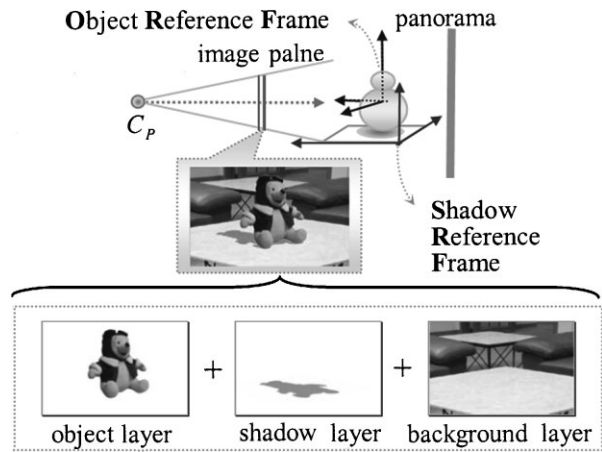


Figure 8. The desired augmented view is composed of three layers of images: the background layer, the shadow layer, and the object layer.

(SRF) in the panorama. It is this reference frame that defines where the shadow of the object is supposed to be projected onto. Once the SRF is determined, the remaining tasks can be divided into three parts: (i) rendering the background layer, (ii) rendering the shadow layer, and (iii) rendering the object layer. An example of the three layers is given in Figure 8. Notice that the SRF should be specified before defining the ORF, because the ORF should be attached to some existing surface in the panorama. Moreover, the shadow can only be rendered after the ORF is determined, because it is a projection of the object whose position depends on the ORF. As to the rendering of background layer, one can easily map the panoramic texture to a 3D model for fast rendering,<sup>2</sup> and hence it will not be described here. In this paper, we assume the camera model (which is used both in image acquisition and rendering of OMs) satisfies the scaled orthographic<sup>17</sup> projection model. Then, the rendering result obtained by our method will be exactly visually consistent. Otherwise, our rendering result of 3D objects and shadows is only a good approximation.

## Selecting a Shadow Reference Frame

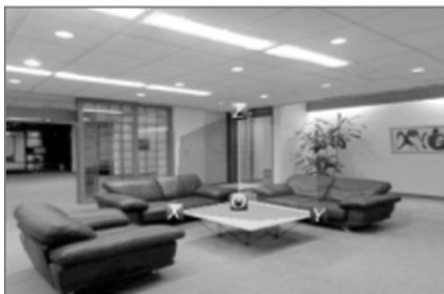
To select a 3D shadow reference in the panorama, we adopt the so-called ‘cuboid-based method’ proposed by Chen *et al.*,<sup>8</sup> which can assist the user to define a 3D reference frame in a 2D image by specifying six vertices of the cuboid projected in the image. By noticing that, for any dewarped view of a panorama, the intrinsic camera parameters (i.e., the image centre and the normalized effective focal length) can be computed directly



(a)



(b)



(c)

Figure 9. An example of selecting the SRF based on a dewarped view of a panorama. (a) shows the initial position of the SRF, (b) and (c) are the convex and concave solution that are related to the illusion of the Necker cuboid.

from the dewarping process, we can further simplify the problem and only require the user to specify four vertices of the cuboid.<sup>18</sup>

Figure 9 illustrates the selection of the SRF. At the beginning, the user is given a dewarped view of the panorama augmented with a dangling SRF, as shown in Figure 9(a). As a convention, the  $x$ - $y$  plane of the SRF is the planar surface to which the inserted object is supposed to attach. Based on the dewarped view, the user can then drag the four endpoints of the reference frame to any image points until the shaded half-cuboid complies with the user's 3D perception, as illustrated in Figure 9(b).

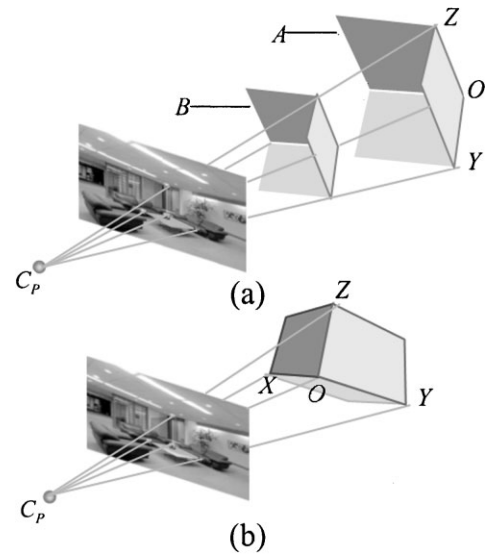


Figure 10. 3D illustration of the two solutions shown in Figure 9: (a) the right-hand-rule case, and (b) the left-hand-rule case. Figure (a) also illustrates that the translation can only be determined up to a scale.

Once the user has specified an SRF in the dewarped panoramic view, our system can use this information to compute the geometric transformation between this SRF and the panorama reference frame (PRF), which is a global reference frame centred at  $C_p$ , up to an unknown scaling factor. This means that the system can determine the rotation matrix but cannot determine the translation vector completely. For example, in Figure 10(a), it cannot tell whether cuboid A or cuboid B is the correct answer. Fortunately, this does not affect our augmentation process since both answers will produce the same image. Therefore, we can simply choose an arbitrary one, e.g., choose the one whose origin  $O$  is 1000 units away from  $C_p$ . Another byproduct of this computation is that we can determine the length of,  $\overline{OX}$ ,  $\overline{OY}$  and  $\overline{OZ}$ , again, up to a scale.

Computing the 3D orientation and location of the cuboid given the four endpoints in the 2D image is referred to as the problem of *camera pose estimation* in the computer vision community. In particular, given a trihedral with the angles between each pair of lines being  $90^\circ$  in the 3D space, this problem can be converted into a simpler problem of solving a second-order polynomial equation.<sup>18</sup> The two solutions for the second-order polynomial equation are related to the illusion of the Necker cuboid. As illustrated in Figure 10, the solution corresponding to the right-hand rule is shown in (a), while the one corresponding to the left-hand rule

is shown in (b). As mentioned above, our system will overlay a shaded half-cuboid on the dewarped panoramic view, so that the user can easily adjust the half-cuboid interactively until it complies with the user's 3D perception. Figure 9(b) shows the default solution, which is obtained by choosing the right-hand rule. Figure 9(c) shows the other solution obtained by using the left-hand rule, which is obviously not visually coherent with the perceived panoramic view.

## Rendering the Object Layer

After the SRF is specified, we can place the ORF at some initial position in the panorama by referring to the SRF. Figure 11 shows the relationship among the PRF, SRF and ORF. Notice that the SRF is referring to the PRF, and the ORF is referring to the SRF. To browse a panorama, the viewer centre is placed at  $C_p$ , the origin of the PRF. Once the ORF is determined, the object-observation direction, pointing from the viewer centre to the object centre, is also determined. We use the object-observation direction to form a texture plate of the OM, which can either be obtained directly from image acquisition or synthesized by the *t*-DBVM (whenever necessary). Then, we place the centre of the texture plate at  $C_o$ , and the normal of the texture plate is pointing toward  $C_p$ . Further, we also consider the image rotation to allow the user to rotate the object arbitrarily with three degrees of freedom. If the author of an augmented panorama would like to change the position or the size of the object he perceives, he can then drag the object around, or scale the object up and down, in an interactive way. More details can be found in our technical report.<sup>19</sup>

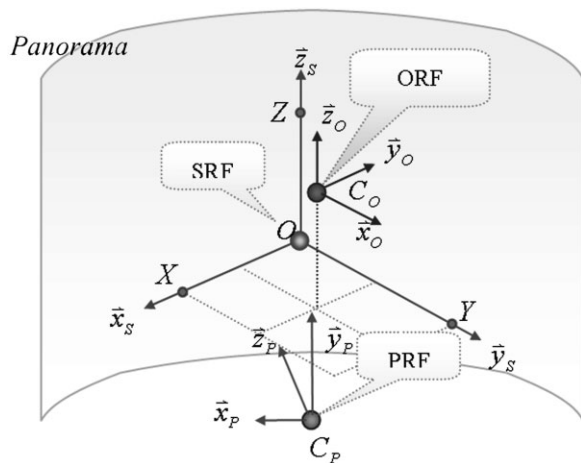


Figure 11. The relationship among the PRF, SRF and ORF.

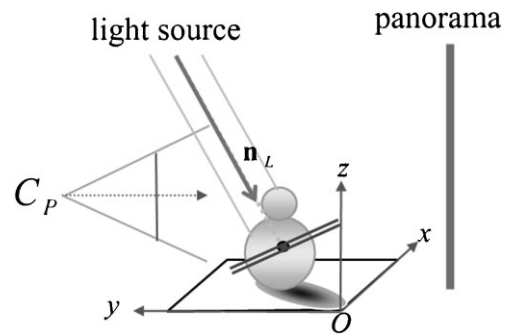


Figure 12. Illustration of casting shadow for an object movie.

## Rendering the Shadow Layer

The next problem is how to generate a realistic shadow of a rotating or translating object without using any 3D geometric model of the object. Our approach is to put the correct shadow map at the correct position with respect to a user-specified SRF. To simplify the problem, we assume that the shadow to be generated is produced by a set of parallel light sources. The lighting directions of those parallel sources can either be estimated from some photographs containing the information of global illumination<sup>20</sup> or specified by the user. As illustrated in Figure 12, we can then retrieve (or 'generate' it by using the *t*-DBVM if necessary) an image of the OM associated with the lighting direction  $\mathbf{n}_L$  (i.e., the image seen from the viewing direction  $\mathbf{n}_L$ ) and use the alpha-channel of this image as the shadow map to cast the shadow on the surfaces where the shadow is chosen to appear, e.g., the *x*-*y* plane of the SRF in most situations. Sometimes, the *y*-*z* or *x*-*z* plane may also be chosen to show shadow, especially when the object is inserted at a place near a wall.

## Experiments

We have implemented a system for inserting OMs into panoramas, which consists of two modules: the *authoring module* and the *browsing module*. With the authoring module, the user can easily insert multiple OMs into any places in a panorama to produce an augmented panorama. Given this augmented panorama, other users can then use the browsing module to look around the panoramic environment and browse the inserted OMs interactively and realistically. (Please refer to <http://smart.iis.sinica.edu.tw/projects/AugPano> for the demo video clips.)



## Authoring Module

To create an augmented panorama containing OMs, the user first inputs a panorama, chooses the places where he wants to insert OMs, and then begins to use the simple interactive manner described above to specify the SRFs for the OMs to be inserted. Figure 13 displays a few snapshots of an authoring example for demonstrating the features of the authoring module. Figure 13(a) shows the specified SRF, where both the  $x$ - $y$  plane (the counter top) and the  $x$ - $z$  plane (the wall) are chosen to display the shadow. Figure 13(b) shows the initial result after putting the Kitty on the counter without any human adjustment. It can be seen that both the position and perspective of the inserted Kitty seem to be already quite reasonable. To view the result more carefully, we can zoom into the augmented image. We can then adjust the position and intensity of the light source to produce a desired shadow. Figure 13(c) shows the augmented image after zooming in and rendering a shadow. Next, we move the object around, and the system will automatically adjust the perspective and the size of the object image. From Figure 13(d), we can see that the system has rendered an image appropriate for this view, whose perspective is different from the one shown in Figure 13(c). If the user scales the object up and down,

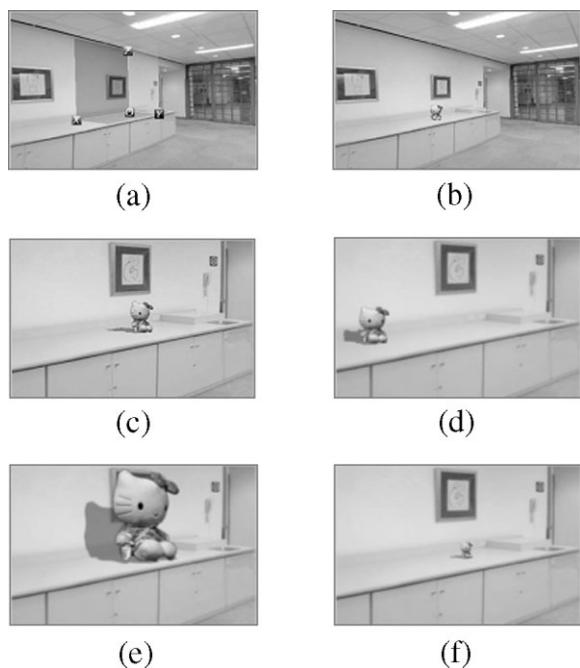


Figure 13. A few snapshots of an example for demonstrating the authoring of an augmented panorama.

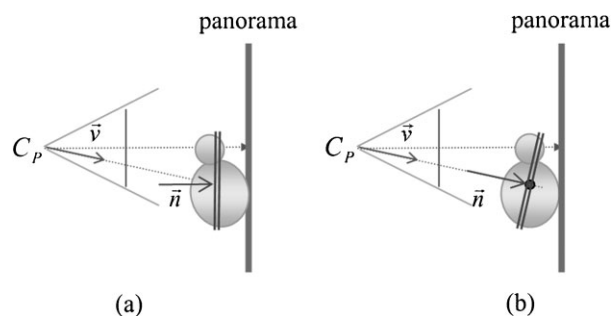


Figure 14. Illustration of the 3D configuration (a) using straightforward manual composition and (b) using our method.

the object can maintain its contact with the counter and the shadow remains visually 3D-consistent, as demonstrated in Figures 13(e) and (f).

Some people may wonder if the integration of the object movie and the panorama can be achieved by manual composition with some image-processing tools, such as Adobe Photoshop. One problem associated with this manual approach is that it will be quite tedious to render the shadow for each image contained in the object movie. Another more serious problem is that, by using the straightforward manual approach, which can be illustrated as in Figure 14(a), the 3D perception of the object may look weird, or geometrically distorted, especially when the object happens to be located away from the centre of the view. On the other hand, our composition method, illustrated in Figure 14(b), can solve this problem easily. Please refer to <http://smart.iis.sinica.edu.tw/projects/AugPano> for a comparison using an example.

## Browsing Module

The browsing module allows the user to view the augmented panorama produced by the authoring module described above. Figure 2 has shown a few snapshots of the augmented results. Among the many problems to be solved, one concerns the difference of image resolution between the panorama and the OMs. Because a panorama needs to cover a larger area than an OM, its resolution is usually much lower than the OM as we zoom in to play the OM. This incompatibility of image resolution may make the panorama look like a separate background from the foreground object. To solve this problem without an overwhelming increase in the memory size of the panorama, we simply took a few closer shots for the places where we were planning

to insert objects (especially for relatively small objects). Since the images taken for displaying closer views and the images taken for producing the panorama are all taken at the same camera nodal point, the relationship between each pair of images is a homography. Also, the relation between a closer view and a dewarped panoramic view is also a homography. Therefore, we can estimate the projection matrix between these two images by specifying four pairs of corresponding points. Once the matrix describing the homography is known, we can smoothly hop from one image to the other in the following way.

Let the wider view of the panoramic environment that the user is currently watching be represented by a pan angle  $\theta_c$ , a tilt angle  $\phi_c$ , and a focal length  $f_c$ , as shown in Figure 15(b). Based on the estimated homography matrix, our authoring system first projects the image centre and the four corners of the narrower-view image onto the dewarped panoramic image and then determines the pan angle  $\theta_n$ , the tilt angle  $\phi_n$  and the focal length  $f_n$  in panoramic environment after zooming into the close view. Next, we can place a textured plate, specified by  $(\theta_n, \phi_n)$  and  $f_n$  and texture-mapped by the close-view image, in front of the panoramic model. During the hopping, we interpolate the pan angle, tilt angle and focal length from  $(\theta_c, \phi_c, f_c)$  to  $(\theta_n, \phi_n, f_n)$  and vary the alpha value from completely transparent to completely opaque. Figure 15(c) shows an intermediate result as we

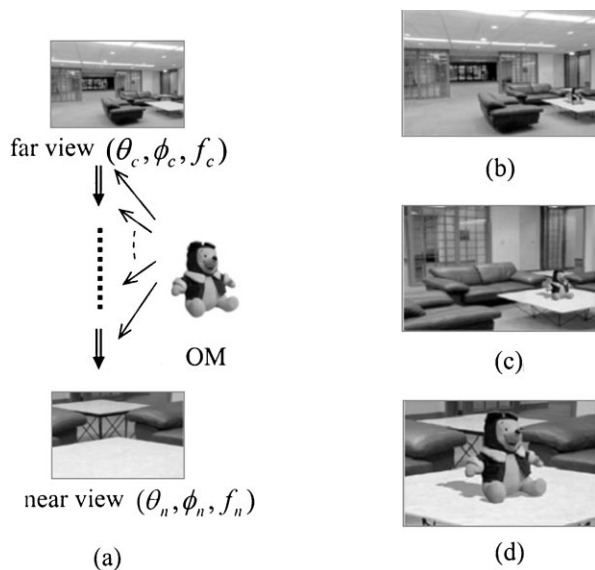


Figure 15. (a) Smooth transition from a farther view to a closer view is desired after OM insertion. (b) Augmented views at  $t=0.0$ , (c) at  $t=0.6$ , and (d) at  $t=1.0$ .

hop from Figure 15(b) to the Figure 15(d). The transition looks very smooth in our experiments.

## Conclusion and Future Work

In this paper, we have introduced an image-based approach to augment a panorama with OMs. A critical issue of this method is how to generate the novel views required for showing an OM in different places of a panorama, and we have proposed a view morphing technique, called *t*-DBVM, to solve this problem. The major characteristic of this paper is that the proposed method is *purely image-based* in the sense that it does not have to reconstruct the 3D geometric model of the object to be inserted into the panorama. That is, it can avoid the tedious 3D reconstruction task while still allowing the user to simultaneously view the panorama and the inserted objects from different perspectives, interactively and with correct 3D perception and shadowing effect. Based on the proposed method, we have implemented an easy-to-use system for authoring and for browsing augmented panoramas. Experimental results have shown that the composite images rendered by our system are visually convincing. Our method can be easily extended to display stereo OMs in a stereo panorama.<sup>21</sup> Also, it can be used to insert objects into a panoramic video if we can track the cuboid structure in the panoramic video.

## References

1. Chen SE. QuickTime VR: an image-based approach to virtual environment navigation. In *Proceedings of Computer Graphics (SIGGRAPH'95)*. ACM Press: Los Angeles, CA, 1995; 29–38.
2. Szeliski R, Shum H-Y. Creating full view panoramic image mosaics and texture-mapped models. In *Proceedings of Computer Graphics (SIGGRAPH'97)*. ACM Press: Los Angeles, CA, 1997; 251–258.
3. Gluckman J, Nayer SK. Ego-motion and omnidirectional cameras. In *Proceedings of Sixth International Conference on Computer Vision (ICCV'98)*. IEEE CS: Bombay, India, 1998; 999–1005.
4. Koller D, Klinker G, Rose E, Breen D, Whitaker R, Tuceryan M. Real-time vision-based camera tracking for augmented reality application. In *Proceedings of ACM Symposium Virtual Reality Software and Technology (VRST '97)*, 1997; 87–94.
5. Schumann H, Burtescu S, Siering F. Applying augmented reality techniques in the field of interactive collaborative design. In *Proceedings of SMILE Workshop 3D Structure from*

- Multiple Images of Large-Scale Environments, in Conjunction with ECCV '98*, 1998.
6. Kutulakos KN, Vallino J. Calibration-free augmented reality. *IEEE Transactions on Visualization and Computer Graphics* 1998; 4(1): 1–20.
  7. Seo Y, Hong KS. Calibration-free augmented reality in perspective. *IEEE Transactions on Visualization and Computer Graphics* 2000; 6(4): 346–358.
  8. Chen CS, Yu CK, Hung YP. New calibration-free approach for augmented reality based on parameterized cuboid structure. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV '99)*. IEEE CS: Corfu, Greece, 1999; 23–37.
  9. Katayama A, Sakagawa Y, Yamamoto H, Tamura H. Shading and shadow-casting in image-based rendering without geometric models. In *Conference Abstracts and Applications (SIGGRAPH '99)*, 1999; 275.
  10. Uchiyama S, Yamamoto H, Katayama A, Tamura H. Presentation and interaction of virtual 3D objects without geometric model. In *Proceedings of the International Conference on Human-Computer Interaction (HCI'97)*, Vol. 2, 1997; 869–872.
  11. Levoy M, Hanrahan P. Light field rendering. In *Proceedings of Computer Graphics (SIGGRAPH'96)*. ACM Press: New Orleans, LA, 1996; 31–42.
  12. Gortler SJ, Grzeszczuk R, Szeliski R, Cohen MF. The lumph. In *Computer Graphics (SIGGRAPH'96)*. ACM Press: New Orleans, LA, 1996; 43–54.
  13. Kutulakos K, Seitz SM. A theory of space carving. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*. IEEE CS: Corfu, Greece, 1999; 307–314.
  14. Seitz SM, Dyer CR. View morphing. In *Proceedings of Computer Graphics (SIGGRAPH '96)*. ACM Press: New Orleans, LA, 1996; 21–30.
  15. Huang HC, Nain SH, Hung YP, Cheng T. Disparity-based view morphing: a new technique for image-based rendering. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST'98)*. Taipei: Taiwan, 1998; 9–16.
  16. Huang HC, Kao CC, Lin YC, Hung YP. Disparity-based view interpolation for multiple-perspective stereoscopic displays. In *Proceedings of SPIE Conference on Stereoscopic Displays and Virtual Reality Systems VII*, San Jose, CA, Vol. 3957, 2000; 102–113.
  17. Mundy JL, Zisserman A. *Geometric Invariance in Computer Vision*. MIT Press: Cambridge, MA, 1992.
  18. Chen CS, Hsieh WT. Composition of 3D graphic objects and panorama. In *Proceedings of 10th International Conference on Artificial Reality and Tele-existence (ICAT2000)*. Taipei: Taiwan, 2000; 207–214.
  19. Hung YP, Chen CS, Tang GY, Tsai YP, Huang CF, Lin SW, Yu CH. *A Purely Image-Based Approach to Augmenting Panoramas with Object Movies*. Technical report TR-IIS-01-011, Institute of Information Science, Academia Sinica, Taipei, Taiwan, 2001.
  20. Debevec P. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Computer Graphics (SIGGRAPH'98)*. ACM Press: Orlando, FL, 1998; 189–198.
  21. Huang HC, Hung YP. Panoramic stereo imaging system with automatic disparity warping and seaming. *Graphical Models and Image Processing* 1998; 60(3): 196–208.