Appearance-Based Virtual View Generation From Multicamera Videos Captured in the 3-D Room

Hideo Saito, *Member, IEEE*, Shigeyuki Baba, and Takeo Kanade, *Fellow, IEEE*

***Abstract*—We present an appearance-based virtual view generation method that allows viewers to fly through a real dynamic scene. The scene is captured by multiple synchronized cameras. Arbitrary views are generated by interpolating two original camera-views near the given viewpoint. The quality of the generated synthetic view is determined by the precision, consistency and density of correspondences between the two images.**

All or most of previous work that uses interpolation extracts the correspondences from these two images. However, not only is it difficult to do so reliably (the task requires a good stereo algorithm), but also the two images alone sometimes do not have enough information, due to problems such as occlusion. Instead, we take advantage of the fact that we have many views, from which we can extract much more reliable and comprehensive three-dimensional (3-D) geometry of the scene as a 3-D model. Dense and precise correspondences between the two images, to be used for interpolation, are obtained using this constructed 3-D model. Pseudo correspondences are even obtained for regions occluded in one of the cameras and then we used to correctly interpolate between the two images. Our method of 3-D modeling from multiple images uses the Multiple Baseline Stereo method and the Shape from Silhouette method. The virtual view sequences are presented for demonstrating the performance of the virtual view generation in the 3-D Room.

***Index Terms*—Image based rendering, model based rendering, multibaseline stereo, multiple-view images, shape from silhouette, 3-D model.**

I. Introduction

METHODS for three-dimensional (3-D) shape reconstruction from multiple-view images have recently received significant research, mainly because of advances in computation power and data handling capacity. Research in 3-D shape reconstruction from multiple-view images has conventionally been applied in robot vision and machine vision systems, in which the reconstructed 3-D shape is used for recognizing the real scene structure and object shape. For those kinds of applications, the 3-D shape itself is the goal of the reconstruction.

New applications of 3-D shape reconstruction have recently been introduced [26], [29]. One of such application is arbitrary view generation from multiple-view images, in which the new views are generated by rendering pixel values of input images in accordance with the geometry of the new view and the 3-D structure model of the scene [1], [14], [18], [19], [20], [28], [32]. The 3-D shape reconstruction techniques can be applied to recover the 3-D model that is used for generating new views. Such a framework for generating new views via recovery of a 3-D model is generally called "model-based rendering (MBR)." MBR has the advantage of handling the occlusion problem as they make use of the 3-D models. However, registration errors of texture mapping onto the constructed 3-D model may cause blur of synthesized virtual images.

Alternatively, image-based rendering (IBR) has recently been developed for generating new views from multiple-view images without recovering the 3-D shape of the object. Because IBR is essentially based on two-dimensional (2–D) image processing (cut, warp, paste, etc.), the errors in 3-D shape reconstruction, such as dense stereo [23], do not affect the quality of the generated new images as much as for model-based rendering.

In this paper, we present a view interpolation approach that we call *Appearance-Based Virtual-View Generation*, which captures the best features of both MBR and IBR. This method generates virtual views taken by multiple camera images of the "3-D Room" [15], which we have developed for digitizing dynamic events, as is and in their entirety. First, a 3-D model of a scene is reconstructed from the multiple images by using "Multiple Baseline Stereo" (MBS) [22] and "Shape from Silhouette" (SS) [4], [25]. Taking advantage of the fact that we have 3-D models of the scene, geometrically accurate correspondences between pairs of images are obtained with the aid of the 3-D model. The precise and dense correspondences are used to generate virtual views at arbitrary viewpoints without losing pixels even in partially occluded regions. With this method, the virtual appearance views are generated in accordance with the correspondence between input images. Even though the image generation procedure is based on a simple 2-D image morphing process, the generated virtual views reasonably represent 3-D structure of the scene because of the 3-D structure information included in the correspondence between the images. The 3-D structure recovery helps to avoid the occlusion problem between the images used to generate virtual views.

We demonstrate the performance of the proposed framework for virtual view generation from multiple cameras by showing several virtual image sequences of a dynamic event.

II. Related Work

Recent research in both computer vision and graphics has made important steps toward generating new views. Such studies for generating new views can be broken down into

Manuscript received October 26, 2000; revised September 9, 2002. The associate editor coordinating the review of this paper and approving it for publication was Dr. Thomas R. Gardos.

H. Saito is with the Department of Information and Computer Science, Keio University, Yokohama, Japan (e-mail: saito@ics.keio.ac.jp).

S. Baba is with the HPS Business Development Pj., Sony Corporation, Tokyo, Japan (e-mail: Shigeyuki.Baba@jp.sony.com).

T. Kanade is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: tk@cs.cmu.edu).

Digital Object Identifier 10.1109/TMM.2003.813283

two basic groups: generating new views by mapping of the texture of input images onto 3-D structure models that are reconstructed from range images (MBR), and generating new views directly from multiple images (IBR).

In the model based rendering, a 3-D structure model is initially reconstructed using volumetric integration of range images. Hilton *et al.* [11], Curless and Levoy [5], and Wheeler *et al.* [35], led to several robust approaches to recovering global 3-D geometry from range images from different view. In most cases, the range images are obtained by a direct range-scanning hardware, which is relatively slow and costly for a dynamic multiple sensor modeling system. To avoid such disadvantages of the range-scanning hardware, multiple-view images are used for the reconstruction of the 3-D models. Debevec *et al.* [6] developed an interactive editing system to recover 3-D structure model of the object and a view-dependent texture mapping scheme to the model [7]. Faugeras *et al.* [8] developed a system which can generate 3-D models of a static environment semi-automatically from sequences of images. Seitz *et al.* [27] proposed a method for coloring visible voxel from multiple views. In this method, each voxel's visibility is checked by using the color consistency of projected points in all the input images, and then the colored visible voxels representing the object shape provide virtual view. Since the color consistency is important, this method is not suitable for multiple camera system in which sensitivity varies in different cameras, but rather suitable for single camera system with turntable of the object. Our group [23], [32] demonstrated automated creation of four-dimensional (4-D) (3-D + time) models for time-varying scenes by applying image-based stereo for generation of range images and volumetric integration of the range images, together with texture mapping and rendering of new views. These methods have the advantage of handling the occlusion problem as they make use of the 3-D models. However, texture mapping onto the constructed 3-D model with errors may cause blur of synthesized virtual images.

Image-based rendering method does not require any 3-D models for synthesizing virtual images. Katayama *et al.* demonstrated that images from a dense set of viewing positions on a plane can be directly used to generate images for arbitrary viewing positions [16]. Levoy and Hanrahan [18] and Gortler *et al.* [10] extend this concept to construct a 4-D field representing all light rays passing through two parallel planes. New view generation is posed as computing the correct 2-D cross section of the field of light rays. A major problem with these approaches is that thousands of real images may be required to generate new views realistically, therefore making the extension to dynamic scene modeling impractical.

View interpolation [3], [34] is one of the first approaches that exploited correspondences between images to project pixels in real images into a virtual image plane. This approach linearly interpolates the correspondences, or flow vectors, to predict intermediate viewpoints. View morphing [28] is an extension of image morphing [2], that correctly handles the 3-D geometry of multiple views. Avidan *et al.* [1] has proposed a geometrically correct way for generating new views from three input images by using the trilinear tensor that has been proposed by them. In those methods, correspondences between the original images must be specified for warping the original images to generate intermediate views. The correspondences are generally given manually [3], [34], [28], or by the use of optical-flow at boundary [1] between two views. In the method presented in our paper, the correspondences are generated from the 3-D structure models reconstructed from multiple input images.

Since our approach to generate new views is based on view interpolation, the virtual viewpoint is not determined by an explicit 3-D specification in the object space, but rather is specified as relative weight factors of interpolating input cameras. This means that the interpolated virtual viewpoint is constrained to be along the segments connecting the input cameras. This is a limitation for viewpoint positioning, but instead we can obtain an advantages of our appearance-based virtual view generating algorithm. In most of the model based rendering methods, the texture rendered onto the model surface is suffered by the error of the recovered 3-D model, that results in degrade of virtual images even if the view point is the same as the real camera position. Contrary to such model based rendering methods, we obtain the full quality image from the same viewpoint as an original camera, by using the view interpolation-based method.

There are other bodies of work involving multiple camera systems. Our group has developed a system using a number of cameras for digitizing whole real world events including 3-D shape information of dynamic scenes [14], [23], [32]. Gavrila *et al.* [9] have developed a multiple camera system for human motion capturing without any sensors on the human body. Jain *et al.* [13] proposed Multiple Perspective Interactive (MPI) Video, which attempts to give viewers control of what they see, by computing 3-D environments for view generation by combining a priori environment models and the dynamic predetermined motion models. In the single camera case, 3-D structure recovery from a moving camera involves using a number of images around the object [24], [30].

Image-based Visual Hull (IBVM) [19] is another virtual view synthesis method from multiple cameras. IN IBVM, the hull shape of the object is represented by the intersection of silhouettes on the epipolar lines of one base camera. Such image-based representation contributes high speed rendering with conventional image rendering hardware. Since the visual hull reconstructed from silhouette images cannot represent the actual 3-D shape, however, it is difficult to render high quality virtual images in case of complicated object shape. IBVM is also difficult to overlay virtual objects with synthesized hull shape, because the explicit 3-D model shape is not represented.

Although the multicamera system is developed to capture dynamic scenes, the proposed algorithm in this paper performs the reconstruction on a frame-by-frame basis, without taking into account the temporal relationship between different frames. Recent research has shown improved reconstruction by using the temporal relationship of the scene [21], [33].

III. The 3-D Room

The "3-D room" is a facility for "4-D" digitization: it captures and models a real time-varying event as a 3-D representation which depends on time (one-dimensional). On the walls and ceiling of the room, a large number of cameras are mounted,

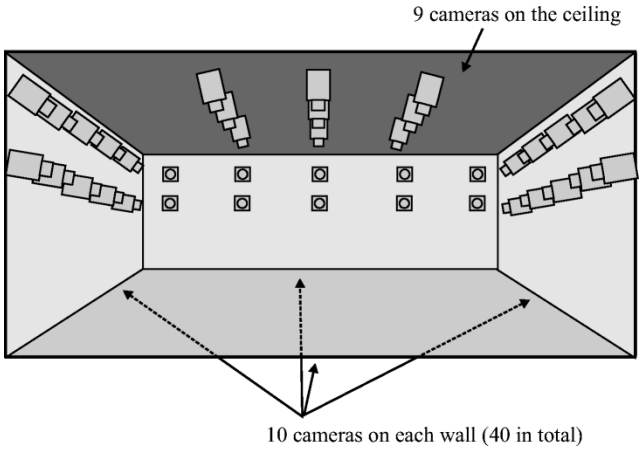

Fig. 1. Camera placement in the 3-D Room.

Fig. 2. Panoramic view of the 3-D Room.

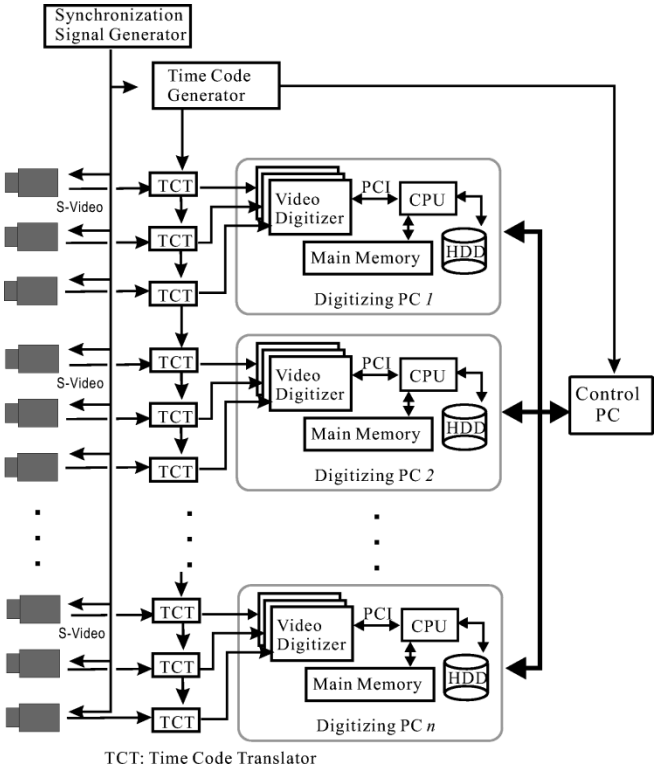

Fig. 3. The digitization system of the 3-D Room consists of 49 synchronized cameras, one time code generator, 49 time code translators, 17 digitizing PCs and one control PC.

all of which are synchronized with a common signal. Our 3-D Room [15] is 20 feet (L) $\times$ 20 feet (W) $\times$ 9 feet (H). As shown in Figs. 1 and 2, 49 cameras are currently distributed inside the room: ten cameras are mounted on each of the four walls, and nine cameras on the ceiling. A PC cluster computer system (currently 17 PCs) can digitize all the video signals from the cameras simultaneously in real time as uncompressed and lossless color images at full video rate ($640 \times 480 \times 2 \times 30$ bytes per seconds). Fig. 3 shows the diagram of the digitization system. The images thus captured are used for generating the virtual view in this paper.

IV. Appearance-Based Virtual View Generation From Multiple Camera Images

Fig. 4 shows the overview of the procedure for generating virtual views from multiple-view image sequences collected in the 3-D Room.

The input image sequences provide depth image sequences by applying multiple baseline stereo frame by frame. The depth images of all cameras are merged into a sequence of 3-D shape models, using a volumetric merging algorithm [5].

For controlling the appearance-based virtual view point in the 3-D Room, two interpolating cameras are selected. An image observed at a different view from any of the actual images can be generated by interpolating between its two spatially adjacent images by using the correspondence between the images. The corresponding points are computed by using the 3-D shape model. The spatial weighting value between the images controls the appearance of the virtual viewpoint.

A. Three-Dimensional Shape Model Reconstruction

Multiple baseline stereo (MBS) [22] is employed to obtain a depth image at every camera in the 3-D Room. Some (2–4) neighboring cameras are selected for computing the MBS of every camera. All the depth images for all cameras are merged to generate a volumetric model. According to the volumetric merging algorithm [5], an implicit function of the 3-D volume of the object shape, which is represented by signed distance to the object surface, is generated from all the depth images. By using the volumetric merging algorithm, errors of the depth images can be averaged in the the implicit surface representation. Then, the implicit surface of the volume is detected by marching cubes algorithm [17], so that the model of the object can be represented by triangle meshes. The number of the meshes of the surface model is finally decreased by mesh simplification algorithm such as QSlim [12].

A volume of interest can be specified by limiting the areas of range images that are merged into the implicit function during the volumetric merging, so that only the objects of interest can be extracted. An example of the reconstructed 3-D shape model in a triangle mesh representation is shown in Fig. 5. This is the view from the ceiling in the motion sequence “A man in the sofa,” in which the man (upper shape) is standing beside the chair (lower shape).

We also employ Shape from Silhouette (SS) for reducing the shape reconstruction error that is caused by wrong estimation of the depth in the MBS execution. In Shape from Silhouette, foreground (silhouette) images are generated for each camera before the computation of 3-D model. Background subtraction

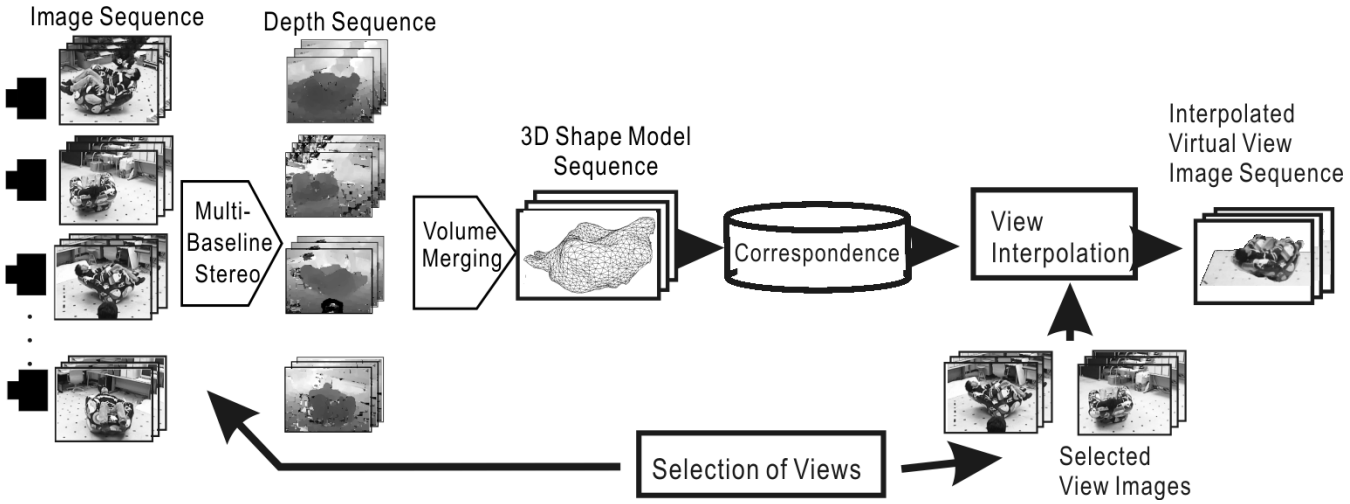

Fig. 4. Overview of the procedure for generating virtual view images from multiple camera in the 3-D Room.

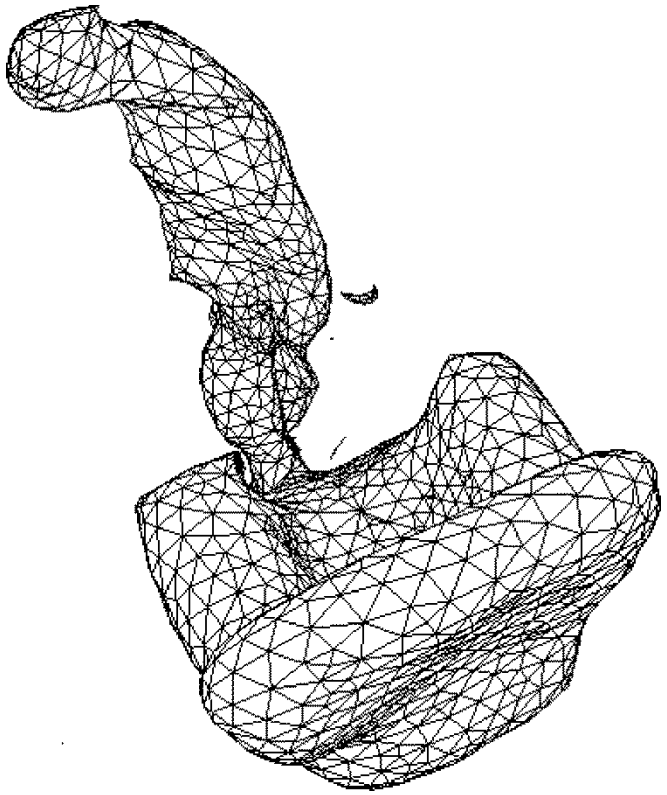

Fig. 5. An example of the reconstructed 3-D shape model, using a triangle mesh representation. The number of triangles in the mesh is 10 000.

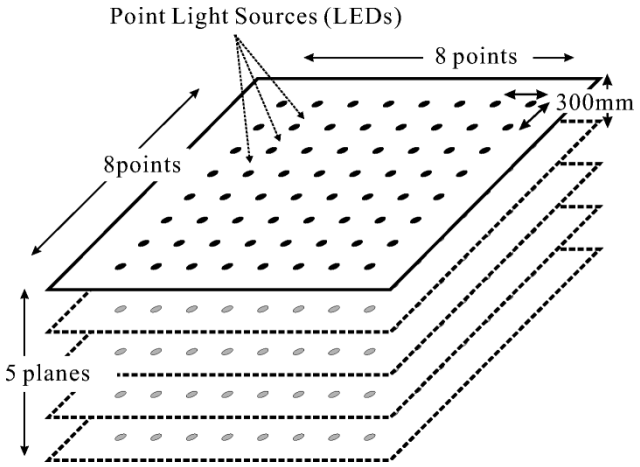

Fig. 6. Placement of point light sources for calibration of every camera in the 3-D Room.

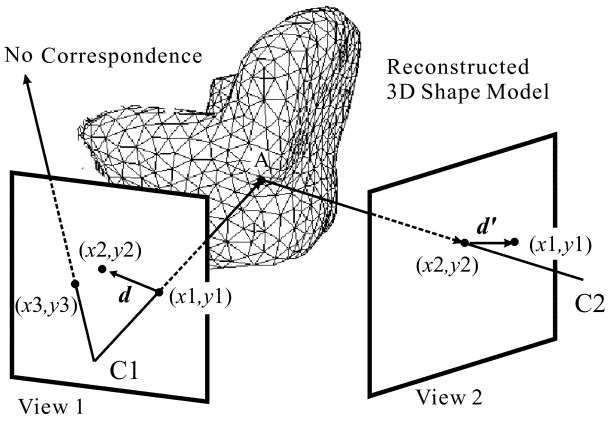

Fig. 7. The scheme for making correspondences in accordance with a 3-D shape model.

is performed for the input images from each of the 49 cameras and dilation and erosion processing are performed to improve the quality of foreground images. After generating foreground images for all cameras, all of the images are back-projected into 3-D space. Each camera viewpoint and its foreground image define a bounding volume. The 3-D model can be reconstructed from intersecting volumes of multiple bounding volumes defined by these foreground images.

For reconstructing the 3-D shape models from multiple images, each camera has to be fully calibrated prior to the reconstruction procedure. We use Tsai's camera calibration method [31], which calculates six degrees-of-freedom of rotation and translation for extrinsic parameters, and five intrinsic parameters which are focal length, aspect ratio of pixel, optical center position, and first order radial lens distortion.

To estimate the camera parameters of all cameras, we place a number of marker point light sources (LEDs) in the volume of interest, and capture images from all cameras. Marker points in

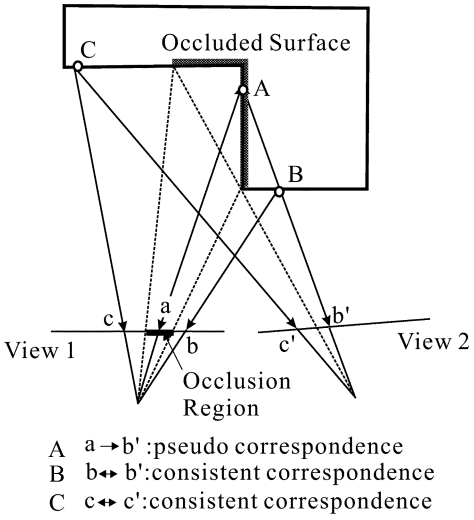

Fig. 8. Consistent correspondence and pseudo correspondence. As the point a is in an occluded region, there is no corresponding point in view B. The pseudo correspondence from the point a is provided by the 3-D shape of the object, by virtually projecting the surface point onto the view B (represented as b'). Point b' is not only the pseudo corresponding point from a, but also the real corresponding point from b in view A.

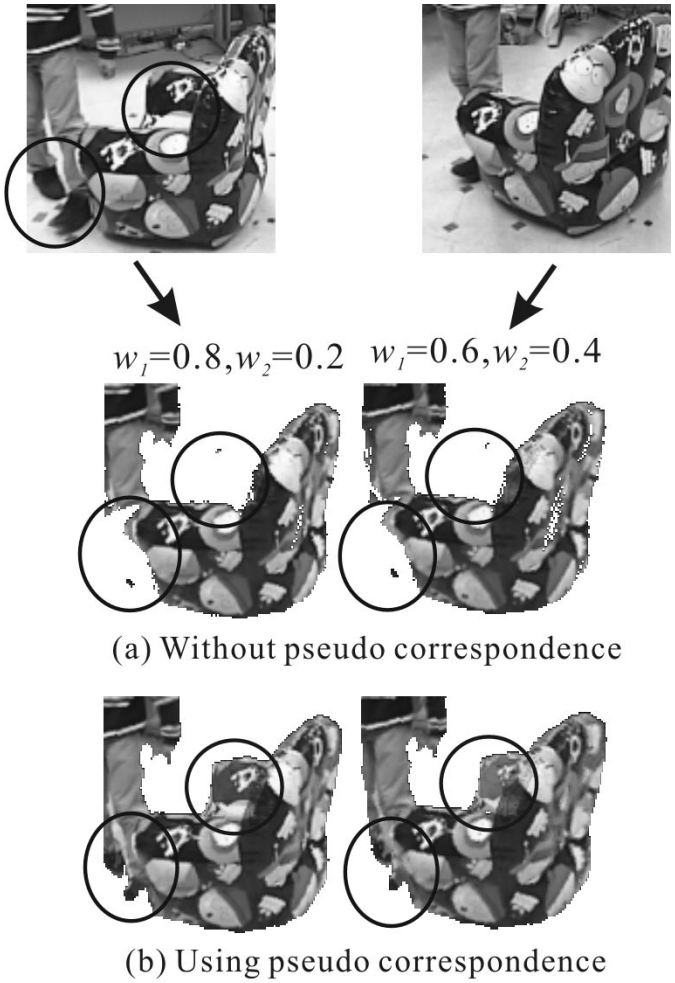

Fig. 9. Effect of the pseudo correspondence in view interpolation. If we only have two views, there is no way to compute the correspondences correctly for the occlusion regions in view 1 (circled areas). Therefore the pixel values in this regions do not exist in (a) the warped image. On the other hand, the 3-D shape model provides the pseudo corresponding points for the occlusion regions in view 1. Thus the pixel values in those regions appear in the (b) interpolated images.

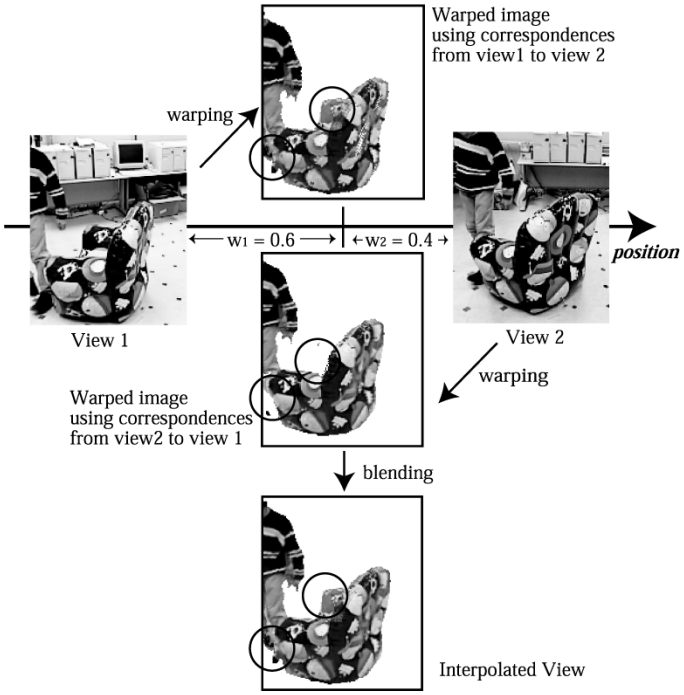

Fig. 10. Interpolation between two views. Each image is warped by the weighted disparity of correspondences. The warped images are then blended for generating the interpolated image.

the volume of interest are shown in Fig. 6, where a plate with 8×8 LEDs at 300 mm intervals is placed at five vertical positions, displaced 300 mm from each other. The images of these point light sources provide the relationship of the 3-D world coordinates to the 2-D image coordinates for every camera. The camera parameters are estimated from this relationship by nonlinear optimization [31].

B. Deriving Pairwise Correspondence From 3-D Model

The 3-D shape model of the object is used to compute correspondences between any pair of views as illustrated in Fig. 7. For a point in view 1, the intersection of the pixel ray with the surface of the 3-D model is computed. The 3-D position of the intersecting point is projected onto the other image, view 2. The projected point is the corresponding point of the pixel in view 1.

In Fig. 7, the ray of the point $(x1, y1)$ intersects the surface at A, and is then projected onto the point $(x2, y2)$. In this case, the point $(x2, y2)$ in view 2 is the corresponding point for the point $(x1, y1)$ in view 1. If there is no intersection on the surface (like the point $(x3, y3)$ in view 1), the pixel does not have any corresponding point.

For each point that has corresponding point, the disparity vector of correspondence is defined. The disparity vector $\mathbf{d}$ for the point $(x1, y1)$ is the flow vector from $(x1, y1)$ to $(x2, y2)$. The disparity vector $\mathbf{d}$' for the point $(x2, y2)$ is the flow vector from $(x2, y2)$ to $(x1, y1)$.

C. Virtual View Generation

1) View Interpolation: Because a virtual view is generated by interpolation of two real neighboring images, the virtual

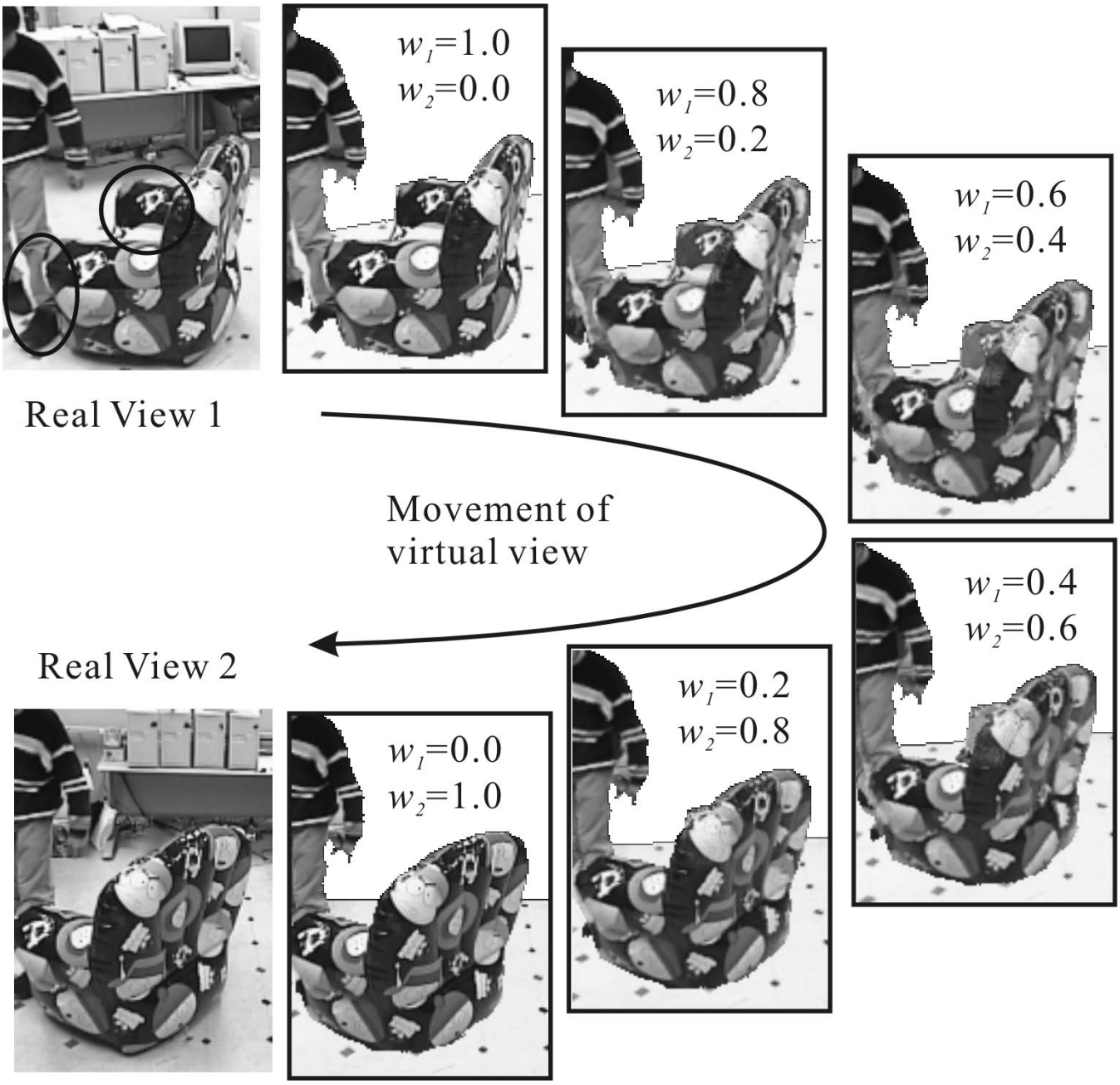

Fig. 11. Generated virtual views by interpolation of two real views using various weighting factors for the fixed instance. The virtual view smoothly moves as the weight factor is changed. The occlusion regions (circled areas) have successfully been interpolated in the virtual views.

viewpoint can not be placed inside the object scene, but can be moved on the lines between real neighboring viewpoints. For controlling the virtual viewpoint in the 3-D Room, two interpolating cameras that are the closest to the virtual viewpoint are selected, and then interpolating weights are determined according to the virtual viewpoint. The intermediate images between the selected two images are generated by interpolation of the selected images from the correspondence between them. The correspondence is computed by using the 3-D shape model as described above. The images are weighted during interpolation in relation to their distance from the virtual viewpoint.

The interpolation is based on the related concepts of "view interpolation" [3] and "view morphing" [2], in which the position and value of every pixel are interpolated from the corresponding points in two images. The following equations are applied to the interpolation:

$$\mathbf{P}_i = w_1\mathbf{P} + w_2\mathbf{P}', \tag{1}$$

$$I_i(\mathbf{P}_i) = w_1 I(\mathbf{P}) + w_2 I'(\mathbf{P}') \tag{2}$$

where

$$w_1 + w_2 = 1$$

$\mathbf{P}$ and $\mathbf{P}$' are the position of the corresponding points in the two views (view 1 and view 2), $I(\mathbf{P})$ and $I'(\mathbf{P}')$ are the pixel values of the corresponding points, and $\mathbf{P}_i$ and $I(\mathbf{P}_i)$ are the interpolated position and pixel value. The interpolation weighting factors are represented by w_1 and w_2, where $w_1 + w_2 = 1$.

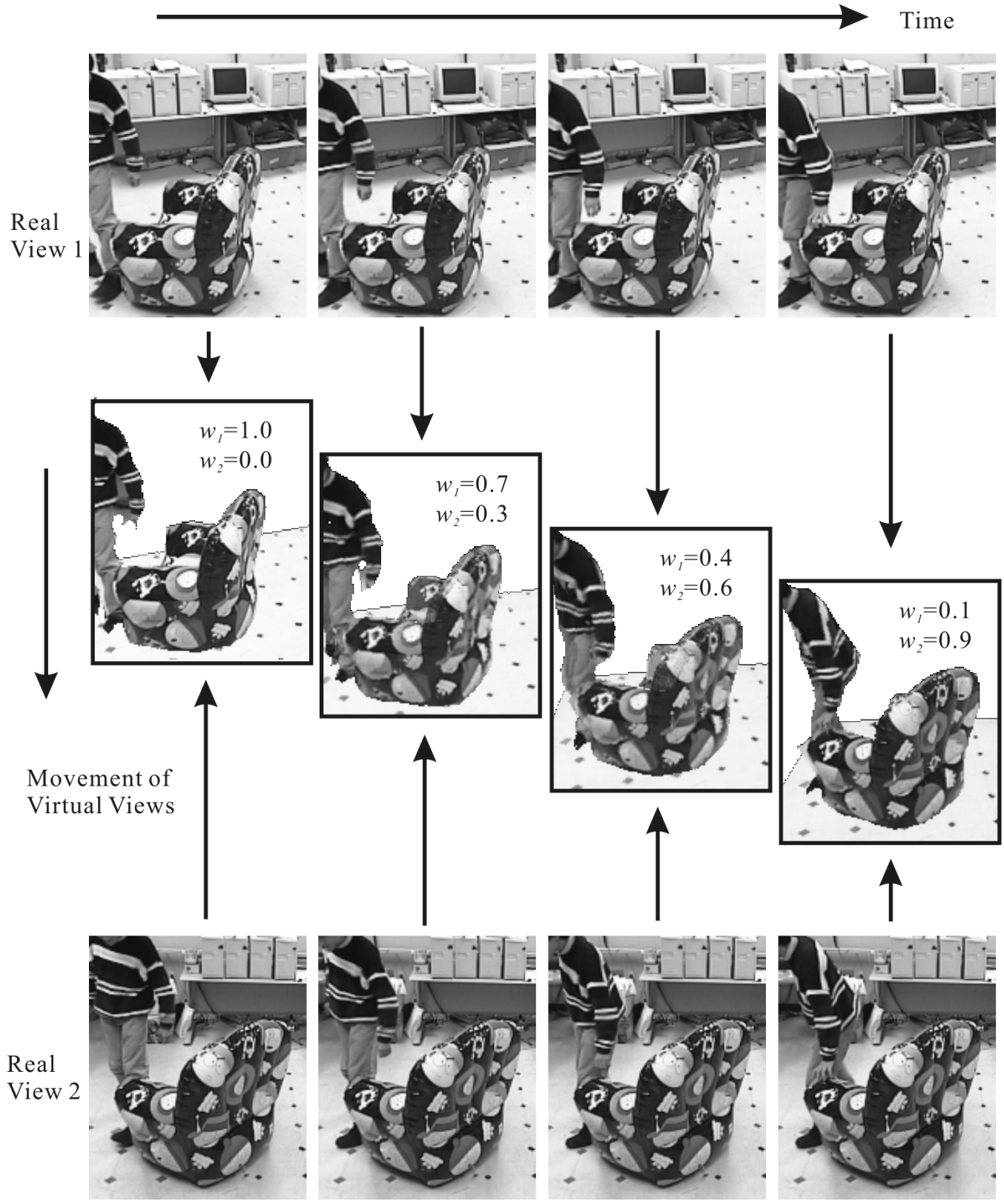

Fig. 12. Generated virtual views for a dynamic event. The real views used for interpolation are also shown with the virtual views. The occlusion region has correctly been interpolated in the image sequence.

2) Pseudo Correspondences for Handling Occlusion: The view interpolation method requires consistent correspondence between two images.

However, it is not unusual that the camera views have occluded regions in the scene. For instance, if we have two cameras and a L shaped object in a scene, as shown in Fig. 8, a part

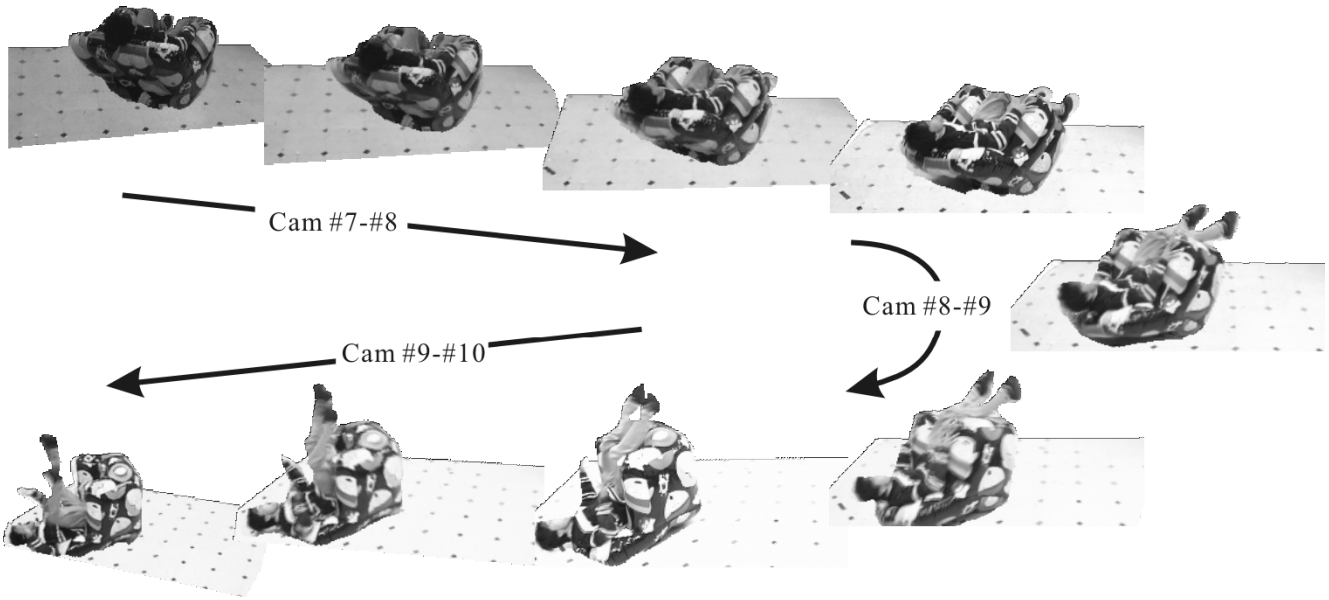

Fig. 13. Generated virtual views for a dynamic event for four cameras (#7, #8, #9, and #10).

of the surface can be in an occlusion region for those views, in which the region can be seen in one image but not in the other.

In this case, interpolation of the pixel value between two views by the method described by (1) and (2) is impossible.

As a result, there is no description of the pixel values in the occlusion region in the generated interpolated images.

To avoid such problems in view interpolation, we introduce the concept of a "pseudo corresponding point" which can be computed for the 3-D shape of the scene. In Fig. 8, a point a in view 1 is reprojected onto b' in view 2 by using the reconstructed shape, even though the point on the object surface cannot be seen in view 2. The point b' is the pseudo corresponding point for a, that corresponds to a only in the geometrical sense. The pseudo corresponding point enables the interpolation of the position by applying the (1) for occluded points. The interpolation of the pixel value is still impossible because the pixel value of the pseudo corresponding point is not actually corresponding in terms of the pixel value of the image. Accordingly, the pixel value is not interpolated for the pseudo correspondence, but just selected to be the pixel value of the occluded point. This is expressed by the following equation.

$$I_i(\mathbf{P}_i) = \begin{cases} I(\mathbf{P}), & \text{if } \mathbf{P} \text{ is not seen in view 2} \\ I'(\mathbf{P}'), & \text{if } \mathbf{P}' \text{ is not seen in view 1.} \end{cases} \quad (3)$$

By using the pseudo corresponding point, we can generate intermediate views without missing parts of occlusion regions.

Pseudo corresponding points can be detected only if the 3-D structure of the scene is available. This suggests that 3-D shape reconstruction plays an important role in view interpolation between two views, even though the interpolation procedure involves only 2-D image processing without 3-D structure.

In Fig. 9, the effect of the pseudo correspondence is presented. If only the two input images are given without any 3-D shape information, the points in the occlusion regions in view 1 (circled areas) cannot have any corresponding points, because no information is available for making the points in the occlusion regions that correspond to the image of the view 2. Therefore, the pixel values in the occlusion region completely vanish in the interpolated images as shown in Fig. 9(a). On the other hand, the complete 3-D shape model, which is reconstructed by the volumetric merging of the depth images at all cameras, enables us to compute the pseudo correspondence even for the occlusion region. Because of the pseudo correspondences, the occlusion region can be successfully interpolated in the virtual view as shown in Fig. 9(b).

D. Algorithm for Virtual View Generation

To apply pseudo correspondences to view interpolation, we first generate the two interpolated images at the same virtual point using the two directed correspondences, from view 1 to view 2 and from view 2 to view 1, separately. Then, the two warped images are blended into a single image in accordance with the interpolation weight. Fig. 10 shows this algorithm.

As described in Section IV-B, disparity vector images $\mathbf{d}(x,y)$ and $\mathbf{d}'(x,y)$ are computed for two interpolating images $I(x,y)$ and $I'(x,y)$ of view 1 and view 2, respectively. For each interpolating image, the warped image is generated by shifting the pixel in the weighted disparity vector. The relation between the warped images $I_w(x,y)$ and $I'_w(x,y)$ and input images $I(x,y)$ and $I'(x,y)$ is

$$\begin{aligned} I_w(x + w_1 d_x(x,y), y + w_1 d_y(x,y)) &= I(x,y), \\ I'_w(x + w_2 d'_x(x,y), y + w_2 d'_y(x,y)) &= I'(x,y) \end{aligned} \quad (4)$$

where

$$\begin{aligned} d_x(x,y) &= x' - x, \\ d_y(x,y) &= y' - y, \\ d'_x(x,y) &= -d_x(x,y), \\ d'_y(x,y) &= -d_y(x,y). \end{aligned} \quad (5)$$

Since the disparity value is not limited to an integer but a floating point value, the shifted pixel can be placed on any point that is not coincident with the pixel sampling point. The value on

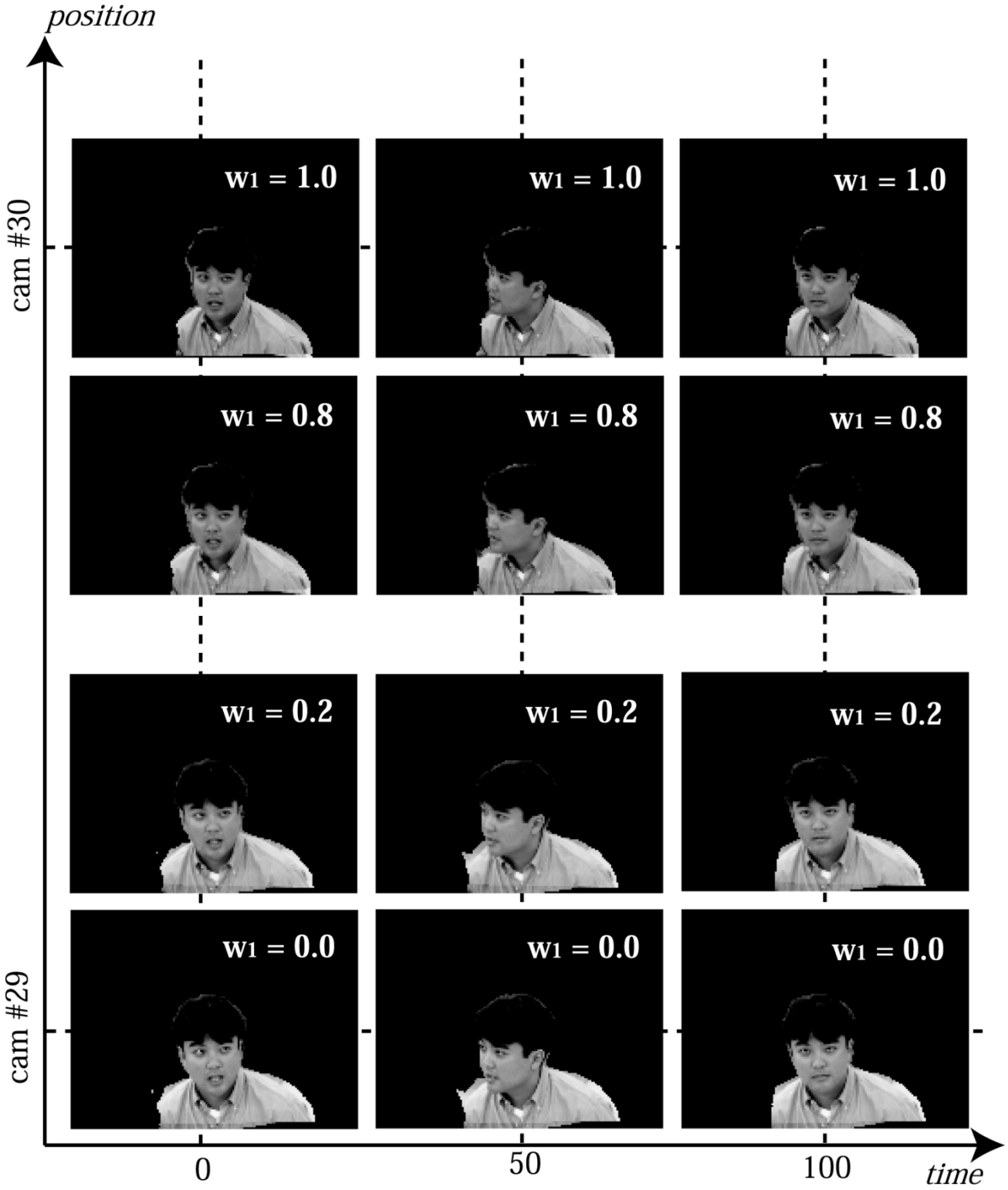

Fig. 14. Example results of appearance based view generation. The horizontal direction represents the frame number of the object event, while the vertical direction represents the virtual camera position. The images between cam #29 and #30 are interpolated by the proposed method.

the pixel point is computed by bilinear interpolation from the neighboring shifted pixel values.

The two warped images are blended into the interpolated image of the two input images according to (6) shown at the bottom of the next page. If a warped pixel value is shifted by the pseudo corresponding point, the pixel value is computed in only one warped image. This corresponds to the first two cases in this equation.

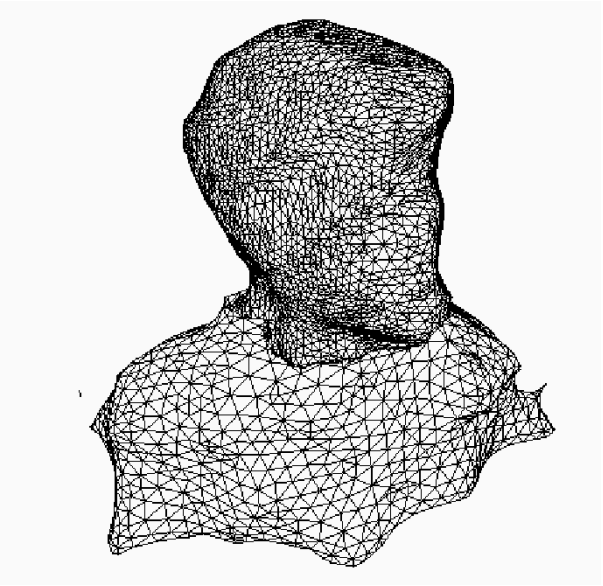

Fig. 15. Three-dimensional shape model for the scene "A man's close-up." Since two groups of cameras with different focal length generate head shape model and body shape model separately, the resolution of two parts is different.

E. Practical Extension of the Algorithm

1) Zooming: We have to account for the fact that focal lengths of various cameras may be different, when dealing with multiple cameras for capturing real views. In this situation, if two neighboring camera-views with different focal lengths are chosen, the object size in the virtual views changes during the movement of viewpoints. If we assume a fixed focal length for the virtual view, the view interpolation described in the previous section cannot work because the focal lengths of the two interpolating images are different. To avoid this problem, it is necessary to add a zooming image feature to the view interpolation.

We modify the view interpolation (2) as follows:

$$\mathbf{P_i} = w_1 \left\{ (\mathbf{P} - \mathbf{C}) \frac{f_v}{f} + \mathbf{C} \right\} + w_2 \left\{ (\mathbf{P'} - \mathbf{C'}) \frac{f_v}{f'} + \mathbf{C'} \right\} \tag{7}$$

$$I_i(\mathbf{P_i}) = w_1 I(\mathbf{P}) + w_2 I'(\mathbf{P'}) \tag{8}$$

where

$$w_1 + w_2 = 1,$$

$\mathbf{C}$ and $\mathbf{C'}$ are the optical centers in view 1 and view 2, respectively. f and f' are the focal lengths of camera 1 (view 1) and camera 2 (view 2). f_v is the focal length of the virtual camera. In this modification, we assume that the focal length affects only the size of the image. Although the focal length affects only the size of the image, the appearance of the virtual camera can zoom in and out in accordance with the focal length f_v. This modification makes the view interpolation method more practical.

2) Viewport Transformation Using Calibration Data: Multiple cameras are usually installed facing toward the center of the object. However, it is difficult to adjust the center of the objects to the exact optical center of each camera-view, even for static objects. If there is an offset between the center of the objects and the optical center in the view, the objects in the virtual view may move out of the field of view during zooming as described in the previous section. To avoid this problem, we transfer the viewport so that the objects can be placed at the center of the virtual view.

Since the calibration data for each camera is computed, we can define the projection matrices using the intrinsic and extrinsic parameters for each camera. Then, if the center of the objects in the world coordinates is defined, it can be projected onto each view using those matrices. Comparing the position of this projected point and the optical center in the views, the transformation value for re-centering objects in views can be computed. Using these transformation values, the center of the objects can be shifted to the optical center in the virtual view.

3) Modified Disparity: By taking into account the zooming and the viewport issues, the disparity calculation shown in Eq. (5) is modified according to the following equations:

$$\begin{aligned} d_x(x,y) &= \left\{ (x' - x'_c)\frac{f_v}{f'} + x'_c \right\} - \left\{ (x - x_c)\frac{f_v}{f} + x_c \right\}, \\ d_y(x,y) &= \left\{ (y' - y'_c)\frac{f_v}{f'} + y'_c \right\} - \left\{ (y - y_c)\frac{f_v}{f} + y_c \right\}, \\ d_x(x,y) &= -d'_x(x,y), d_y(x,y) = -d'_y(x,y) \end{aligned} \tag{9}$$

where f and f' are the focal lengths of view 1 and view 2, respectively, f_v is the focal length of the virtual view, and (x_c, y_c) and (x'_c, y'_c) are the optical centers in view 1 and view 2, respectively.

The modification of disparity enables us to generate two warped images in which the focal length of the virtual camera is fixed to f_v and the center of the objects is aligned with the optical center of the virtual camera.

V. Experiment

A. Virtual View Generation With Various Weighting Factors

The appearance-based virtual views using various weighting factors at a fixed instance for the scene "A man in the sofa" are shown in Fig. 11. This demonstrates that the virtual view smoothly moves as the weight factor is changed. As can be seen,

$$I_i(u,v) = \begin{cases} I_w(x,y), & \text{if } I_w(x,y) \neq 0 \text{ and } I'_w(x,y) = 0, \\ I'_w(x,y), & \text{if } I_w(x,y) = 0 \text{ and } I'_w(x,y) \neq 0, \\ w_1 I(x,y) + w_2 I'(x,y), & \text{otherwise.} \end{cases} \tag{6}$$

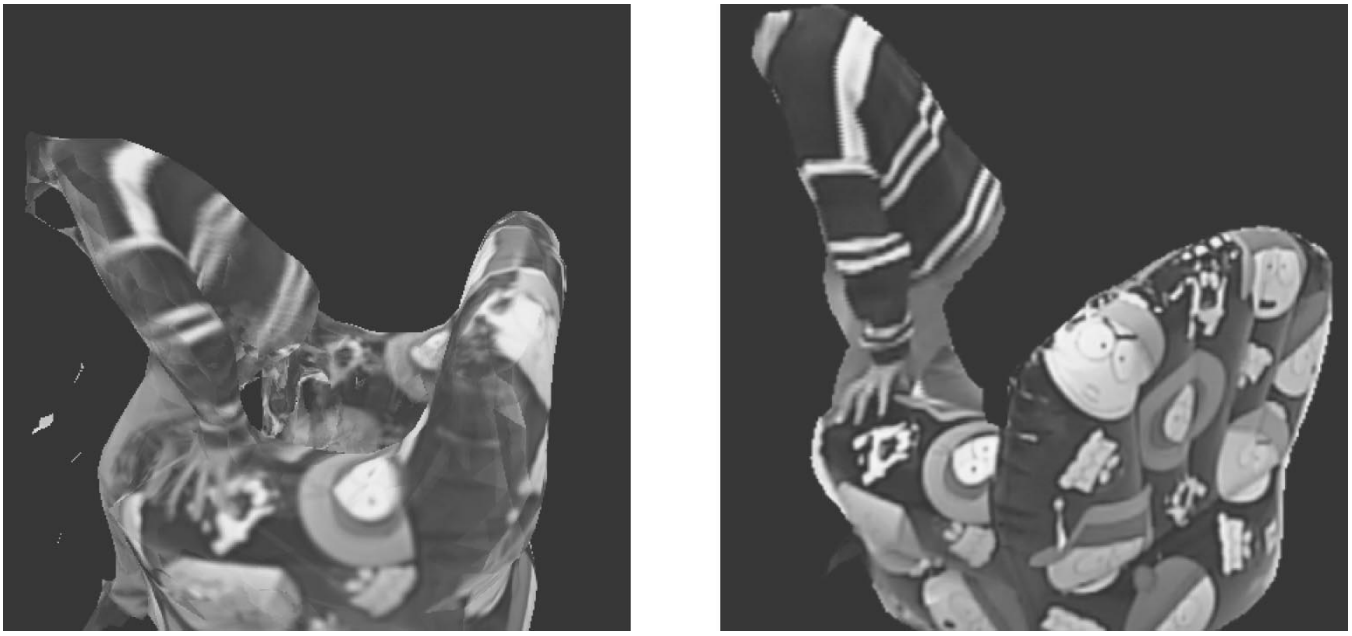

Fig. 16. Comparison between the model based rendering and our appearance-based view virtual view generating method.

the occlusion regions have successfully been interpolated in the virtual images. This demonstrates one of the advantages in using a 3-D model for deriving correspondence between two interpolating images.

Fig. 12 shows the virtual views at several instances. The real views used for interpolation are also shown with the virtual views. This figure also demonstrates that the occlusion region has correctly been interpolated in the image sequence.

Fig. 13 shows an example of an image sequence flying through a virtual view for the dynamic event. As seen in this figure, the virtual viewpoint can move between multiple pairs of cameras for generating the long trajectory of the virtual view.

Fig. 14 shows another example result of appearance-based virtual views for the scene "A man's close-up." In this example, 14 cameras of the 3-D Room capture a close-up view of the object head, while other cameras capture the upper body of the object. The 14 close-up view cameras are distributed on the wall, so some combinations of two neighboring cameras have different focal lengths in this case. In 3-D model reconstruction, the head shape and body shape are separately reconstructed from the group of the close-up view cameras and the group of the other cameras, respectively. This is because the combination of stereo cameras with different focal length fails to generate a satisfactory depth map. After generating a 3-D model separately, two parts are merged into one model which is shown in Fig. 15 for example. As shown in this model, the mesh resolution of two parts is different.

For the scene "A man's close-up," 12 interpolated images whose weighting factors are 0.0, 0.2, 0.8, and 1.0 are generated from the original images of the two cameras (cam #29 and cam #30) at three different time frames (0, 50, and 100). If the weighting factor is either 0.0 or 1.0, the quality of the interpolated images is the same as the original images of the camera #29 or the camera #30. As a result, when we view the scene from the same viewpoint as an original camera, we obtain the full quality image. This is one of the advantages of image interpolation-based new view generation techniques including our method. In most of the model based rendering methods, the texture rendered onto the model surface is blurred by the error of the recovered 3-D model, that results in blurred virtual views even if the view point is the same as the real camera position. Fig. 16 shows a comparison between the model based rendering and our appearance-based virtual view generating method. The image rendered by model-based approach is blurred because of the inaccurate reconstruction of the 3-D model, while the virtual view by our new method is not as blurred by taking advantage of the image-based interpolation-based approach. In the proposed method, the interpolation-based method can be performed even if there is occlusion, because pseudo correspondences for the occlusion area can be obtained using the reconstructed 3-D model.

As shown in Fig. 18, we have developed a GUI-based viewer application, for viewing virtual views which are synthesized by using our methods. With this viewer, users can easily specify the virtual camera position using a mouse and fly through a real dynamic scene.

B. Virtual View Generation From Camera-Views With Different Focal Lengths

Fig. 17 shows another example of the appearance-based view interpolation. In this example, the original views are taken from two cameras with different focal lengths. In order to avoid changing the image size of the virtual viewpoints, we use the fixed focal length f_v for the virtual camera. Using the algorithm described in Section IV-E, the interpolated views are generated with same focal length and the objects in those views are successfully centered using the camera calibration data.

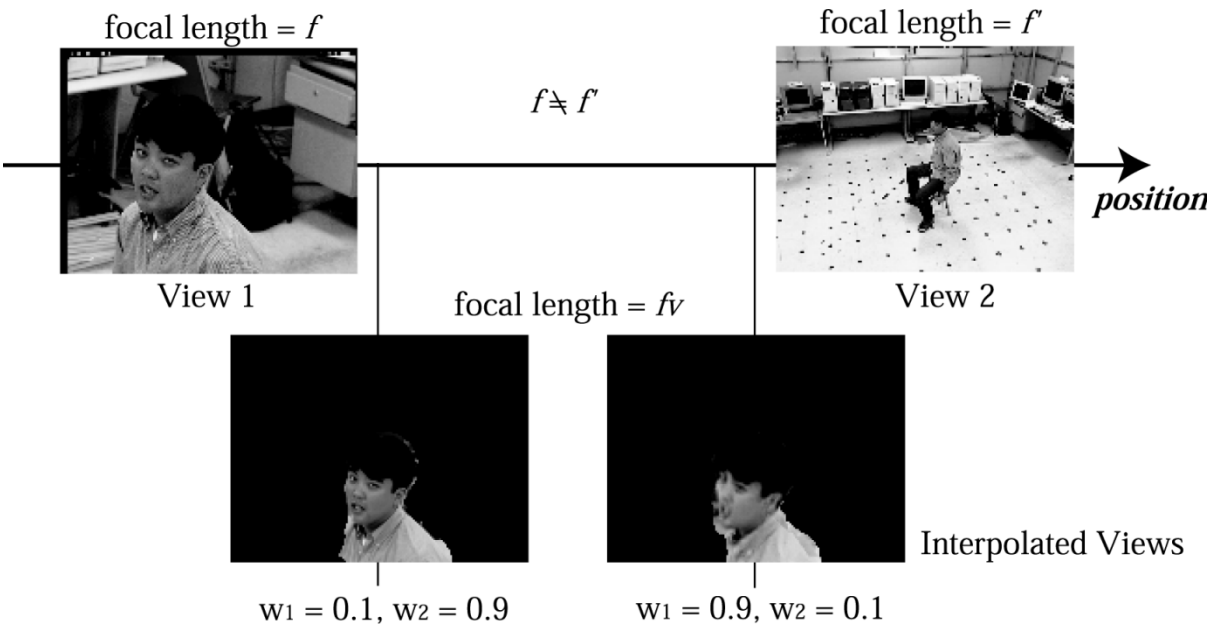

Fig. 17. Example results of appearance based view generation, using two cameras with different focal length.

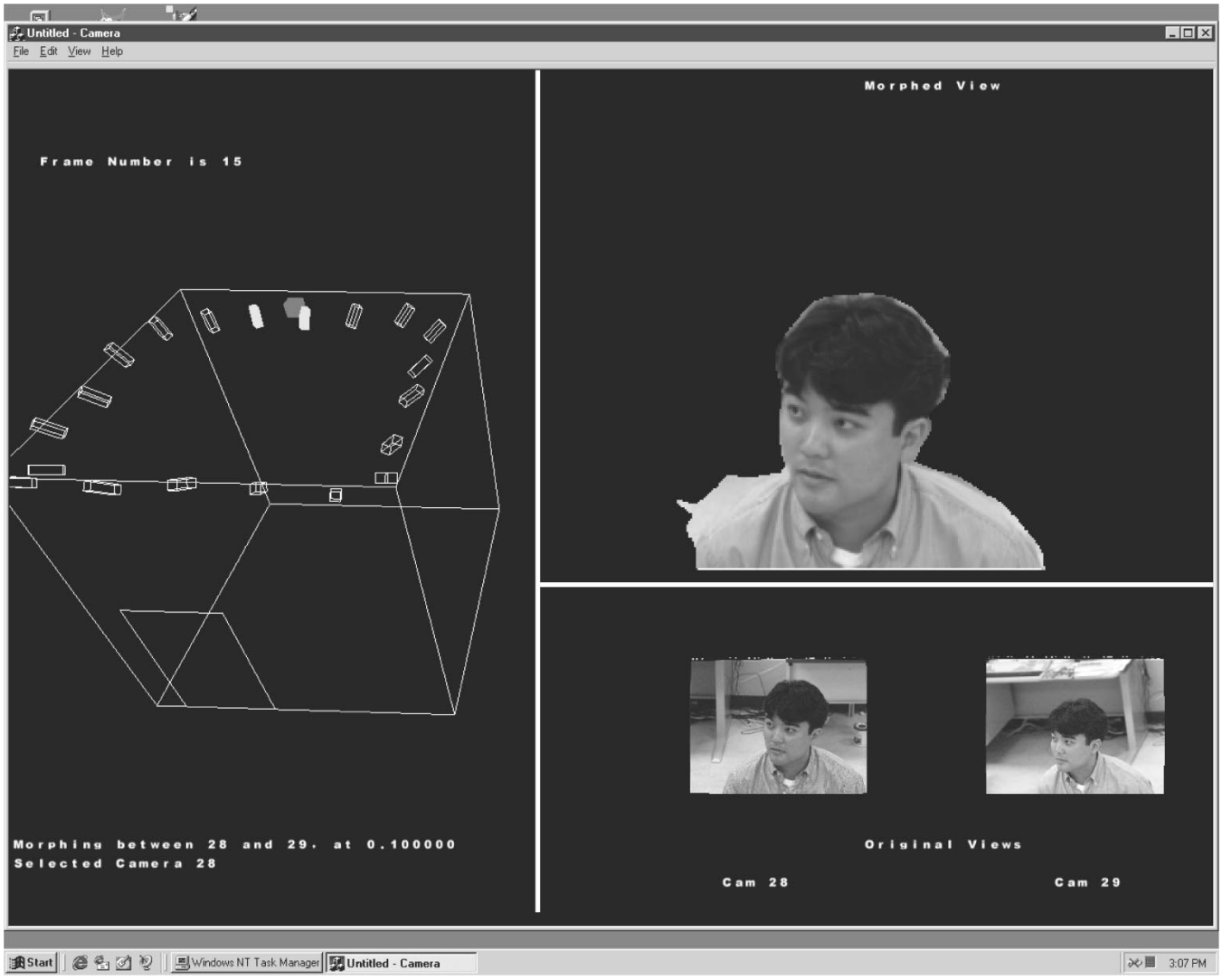

Fig. 18. GUI for displaying the virtual views. In this GUI, the drawing of the 3-D Room displays the arrangement of the cameras, selected interpolating cameras, and virtual view point. The selected images are shown at the bottom right, and generated virtual view is shown at the top right.

VI. Conclusions and Future Work

We have presented a method for virtual view generation from multiple image sequences, which we call the Appearance-Based Virtual-View Generation of dynamic events. First a 3-D volumetric model is recovered from multiple input views. This 3-D model is then used to obtain correspondence between pairs of input images enabling intermediate virtual views to be generated by interpolation. We have defined *Pseudo Correspondences* in order to avoid the occlusion problems. Since our correspondences contain geometric information, virtual views are generated at arbitrary viewpoints without losing pixels even in occlusion regions. Virtual view generation based on Image Based Rendering can be implemented using simple and fast 2-D image processing techniques. That is, once the correspondences are derived from the 3-D model, processing time of the virtual view generation does not depend on complexity of the 3-D objects like other image based rendering methods. Zooming and centering features are also implemented by using the transformation of the disparity vectors and the viewport. Thus the Appearance-Based Virtual-View Generation combines both accuracy and flexibility in the creation of virtual worlds from real views.

In the present method, we do not take into account the geometrical correctness of the interpolated virtual view because we currently only use simple correspondences between images. However, as Seitz *et al.* [28] pointed out in view morphing, such simple correspondence interpolation does not correctly interpolate the geometry of the views. For more realistic new view generation, such correctness of the geometry has to be considered also.

We currently interpolate new views from two views. This means that the virtual camera can only move on the line between the views. We plan to extend our framework to the interpolation of three camera views to make the virtual view move on the plane of these three cameras.

References

[1] S. Avidan and A. Shashua, "Novel view synthesis by cascading trilinear tensors," *IEEE Trans. Vis. Comput. Graph.*, vol. 4, no. 4, pp. 293–306, 1998.

[2] T. Beier and S. Neely, "Feature-based image metamorphosis," in *Proc. SIGGRAPH'92*, 1992, pp. 35–42.

[3] S. Chen and L. Williams, "View interpolation for image synthesis," in *Proc. SIGGRAPH'93*, 1993, pp. 279–288.

[4] C. H. Chein and J. K. Aggarawal, "Identification of 3D objects from multiple silhouettes using quadtrees/octrees," *Comput. Vis., Graph. Image Process.*, ser. 36, pp. 100–113, 1986.

[5] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. SIGGRAPH'96*, 1996, pp. 303–312.

[6] P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *SIGGRAPH'96*, 1996, pp. 11–20.

[7] P. Debevec, G. Borshukov, and Y. Yu, "Efficient view-dependent image-based rendering with projective texture-mapping," in *9th Eurographics Rendering Workshop*, Vienna, Austria, June 1998, pp. 105–116.

[8] O. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller, "3-D Reconstruction of Urban Scenes From Sequences of Images," INRIA, Tech. Rep. 2572, 1995.

[9] D. M. Gavrila and L. S. Davis, "3-D model based tracking of humans in action: Multi-view approach," in *Proc. CVPR'96*, 1996, pp. 73–80.

[10] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH'96*, 1996, pp. 43–54.

[11] A. Hilton, J. Stoddart, J. Illingworth, and T. Windeatt, "Reliable surface reconstruction from multiple range images," in *Proc. ECCV'96*, 1996, pp. 117–126.

[12] P. Heckbert and M. Garland, "Optimal triangulation and quadric-based surface simplification," *J. Comput. Geom.: Theory Applicat.*, vol. 14, pp. 49–65, 1999.

[13] R. Jain and K. Wakimoto, "Multiple perspective interactive video," *Proc. IEEE Conf. on Multimedia Systems*, pp. 202–211, 1995.

[14] T. Kanade, P. W. Rander, and P. J. Narayanan, "Virtualized reality: Constructing virtual worlds from real scenes," *IEEE Multimedia*, vol. 4, no. 1, pp. 34–47, 1997.

[15] T. Kanade, H. Saito, and S. Vedula, "The 3D Room: Digitizing Time-Varying 3D Events by Synchronized Multiple Video Streams," Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-98-34, 1998.

[16] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura, "A view point dependent stereoscopic display using interpolation of multi-viewpoint images," in *Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems II*, vol. 2409, 1995, pp. 11–20.

[17] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface reconstruction algorithm," *Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.

[18] M. Levoy and P. Hanrahan, "Light field rendering," in *SIGGRAPH'96*, 1996, pp. 31–42.

[19] W. Matusik, C. Buehler, R. Raskar, S. Gorlter, and L. McMillan, "Image-based visual hulls," in *Proc. SIGGRAPH 2000*, 2000, pp. 369–374.

[20] S. Moezzi, L. C. Tai, and P. Gerard, "Virtual view generation for 3D digital video," *IEEE MultiMedia*, vol. 4, no. 1, pp. 18–26, 1997.

[21] J. Neumann and Y. Aloimonos, "Spatio-temporal stereo using multi-resolution subdivision surfaces," *Int. J. Comput. Vis.*, vol. 47, no. 1–3, pp. 181–193, Apr.–June 2002.

[22] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 353–363, Apr. 1993.

[23] P. J. Narayanan, P. W. Rander, and T. Kanade, "Constructing virtual worlds using dense stereo," in *ICCV'98*, 1998, pp. 3–10.

[24] M. Pollefeys, R. Koch, and L. V. Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in *ICCV'98*, 1998, pp. 90–95.

[25] M. Potmesil, "Generating octree models of 3D objects from their silhouettes in a sequence of images," *Comput. Vis., Graph. Image Process.*, ser. 40, pp. 277–283, 1987.

[26] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: A unified approach to image-based modeling and spatially immersive displays," in *SIGGRAPH 98*, 1998, pp. 179–188.

[27] S. M. Seitz and C. R. Dyer, "Photorealistic scene reconstruction by voxel coloring," in *Proc. CVPR'97*, 1997, pp. 1067–1073.

[28] ——, "View morphing," in *Proc. SIGGRAPH'96*, 1996, pp. 21–30.

[29] H. Tamura, H. Yamamoto, and A. Katayama, "Mixed reality: Future dreams seen at the border between real and virtual worlds," *IEEE Comput. Graph. Applicat.*, vol. 21, no. 6, pp. 64–70, 2001.

[30] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, 1992.

[31] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE J. Robot. Automat.*, vol. RA-3, no. 4, pp. 323–344, 1987.

[32] S. Vedula, P. W. Rander, H. Saito, and T. Kanade, "Modeling, combining, and rendering dynamic real-world events from image sequences," in *Proc. 4th Conf. Virtual Systems and MultiMedia*, vol. 1, 1998, pp. 326–332.

[33] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Proc. ICCV'99*, vol. 2, Sept. 1999, pp. 722–729.

[34] T. Werner, R. D. Hersch, and V. Hlavac, "Rendering real-world objects using view interpolation," in *Proc. ICCV'95*, 1995, pp. 957–962.

[35] M. D. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus surfaces for modeling 3D objects from multiple range images," in *DARPA Image Understanding Workshop*, 1997, pp. 1229–1236.

Hideo Saito (M'88) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1987, 1989, and 1992, respectively.

He has been on the faculty of Department of Electrical Engineering, Keio University, since 1992. From 1997 until 1999, he was a Visiting Researcher with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. Since 2001, he has been an Associate Professor in the Department of Information and Computer Science, Keio University. Since 2000, he has also been a Researcher with Precursory Research for Embryonic Science and Technology (PRESTO), Japan Science and Technology Corporation (JST), Tokyo, Japan. He has been engaging in the research areas of computer vision, image processing, and human-computer interaction.

Dr. Saito is a member of The Institute of Electronics, Information and Communication Engineers (IEICE), Information Processing Society Japan (IPSJ), The Society of Instrument and Control Engineers (SICE), and The Virtual Reality Society of Japan (VRSJ).

Shigeyuki Baba received B.E. and M.E. degrees in electrical engineering from Waseda University, Tokyo, Japan, in 1992 and 1994, respectively.

He has worked in the Research Center at Sony Corporation, Tokyo, Japan, since 1994. Since 1998 until 2000, he was a Visiting Researcher with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. His work has primarily focused on the research area of computer vision and display systems.

Mr. Baba is a member of the IEICE of Japan.

Takeo Kanade (M'80–SM'88–F'92) received the Ph.D. degree in electrical engineering from Kyoto University, Kyoto, Japan, in 1974.

He is the U.A. Helen Whitaker University Professor of Computer Science and Robotics at Carnegie Mellon University (CMU), Pittsburgh, PA. After holding a faculty position at Department of Information Science, Kyoto University, he joined CMU in 1980, where he was the Director of the Robotics Institute from 1992 to 2001. He has worked in multiple areas of robotics: computer vision, multimedia, manipulators, autonomous mobile robots, and sensors. He has written more than 250 technical papers and reports in these areas, as well as more than 15 patents. He has been the principal investigator of a dozen major vision and robotics projects at CMU. He is the former founding editor of the *International Journal of Computer Vision*.

Dr. Kanade has been elected to the National Academy of Engineering. He is a Fellow of the ACM and the American Association of Artificial Intelligence (AAAI).