

Mixed Reality: Merging Real and Virtual Worlds
Edited by Yuichi Ohta and Hideyuki Tamura
1999 Springer-Verlag

Demonstra-
48), Freder-
ors; (2) the
if Advanced
r); (3) "Sci-
sualization"
19), Center
own Univer-
lenry Fuchs
dical Image
phen Pizer,
uchs, Prin-

colleagues:
MD, Vern
n, Anthony
ck Rolland,
particular
e.

p.506-508,

Fal Joint

ey: A case
295, 1998.

Realizing
16, August

e interac-
Structure,

devices,"

ronments,

The office
tially im-
24, 1998.

Chapter 3

Virtualized Reality: Digitizing a 3D Time-Varying Event As Is and in Real Time

Takeo Kanade
Peter Rander
Sundar Vedula
Hideo Saito

Carnegie Mellon University, U.S.A.

3.1 Introduction

Virtualized Reality is for 4D digitization - capturing and modeling a time-varying 3D event into a computer. A real event is observed by multiple cameras. The multiple synchronized output video streams are analyzed to produce voxel representations of a scene for each moment. First, image-based stereo is used to compute a range map corresponding to each camera view; thus a single time instant of the dynamic event is modeled as a collection of color-range image pairs from different viewpoints, and the full event is modeled as their sequence. These range images are then fused into a global 3D model - consisting of both voxel and surface representations. Finally, as an appearance model, view-dependent texture information is attached by back-projecting the original color images onto the recovered surfaces.

Unlike other techniques, what is captured and modeled in Virtualized Reality is not a collection of views, but an explicit representation of the "event". The

applications of Virtualized Reality include simulation, training, telepresence, and entertainment. For example, by viewing a dynamic event model of a skilled surgeon performing an operation, students could revisit the operation, *free* to observe from anywhere in the reconstructed operating room. Spectators could watch a basketball game from any stationary or moving point on or off the *court*. Once modeled, however, many more things are possible, which are not possible in solely view-based approaches. The recovered scene geometry enables the editing - addition, removal, or alteration - of the event. That is, it is possible to manipulate the reality in the computer; the motion of objects might be altered by computing the Newton-Euler dynamic motion equations after the event, and virtual objects can be added to the event or real components *can* be removed from it at view time.

We first introduced the concept of Virtualized Reality in 1993 [1] [2], based on the work of the **CMU** video-rate stereo machine [3]. Then we moved on to develop the "3D Dome", consisting of 51 cameras and analog VCRs [4], with which we demonstrated an off-line system for Virtualized Reality [5]-[7]. Now, we have developed a new fully digital "3D Room" [8]. This paper presents an overview of the current status - methods and examples - of Virtualized Reality at CMU Robotics Institute. The technical details are presented *in* the above papers.

3.2 Modeling Real Events into Virtual Reality

In the real world, we experiment and observe the results by altering the spatial positions of objects, exerting forces, and adding more objects. Very often, one wants to know the outcome before actually exercising those changes to the real world - remodeling one's house is a **good** example. In *some cases*, real experiments may be dangerous, costly or even impossible *to* perform. In these *cases*, a virtual reality experiment is a convenient alternative. In virtual reality, experimental choices **are** tested in "simulation", and the results are presented mostly by rendering visual, audio and haptic information to the user (**See** Figure 3.1). One of the most important, and yet, least developed capability in this scheme is that of modeling the **real** event for incorporation into virtual reality. While modeling the reality involves many diverse **aspects**, such as geometrical, material, optical, dynamic, and so on, our focus in this paper is spatial and appearance modeling.

Our goal is *to* digitize and model a time-varying three-dimensional event of a large scale in its totality. There are a few aspects that separate our current work from most of the past work. First, the task is 4D digitization - a time sequence of three-dimensional shapes and positions is *to be acquired*. Second, while the 3D sensors provide 2-1/2-D representations of the scene **from each** view, the desired output must be the **full 3D, or** whole scene representation **of** the event - in other words, the set of view dependent data must be **converted** to a scene-centered description. Finally, rather than modeling a **small toy** object on a turn table, **our interest** is modeling an event whose spatial extent is at least that of a room containing multiple people. A large physical space **poses** many issues in calibration, visibility, and illumination.

We **set out** *to* design and develop a dynamic scene modeling system that would realize this vision. First, the video capture **system** must achieve sustained real-time performance in order to capture meaningful events. Second, the observation system

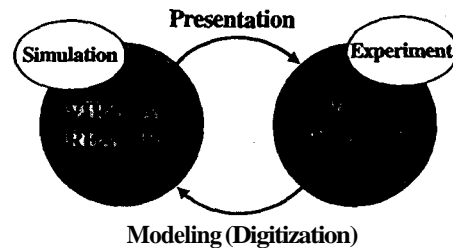


Figure 3.1 Modeling is a critical step for use of virtual reality for simulating on a real world event.

must not interfere with the normal appearance or activity of the event, since the goal is not only geometric modeling, but also modeling of visual appearance. Third, and most importantly, the system must work with minimal human-operator intervention, if at all. The tremendous amount of data generated by multiple video cameras make most of human-interactive approaches unrealistic.

3.3 Related Work

Recent research in both computer vision and graphics has made important steps toward realizing this goal. Work on 3D modeling (e.g., Hilton et al [9], Curless and Levoy [10], Rander et al. [6], and Wheeler et al. [11]) presents volumetric integration of range images for recovering global 3D geometry. Sato et al. [12] have developed techniques to model object reflectance as well. Note that most of these techniques rely on direct range-scanning hardware, which tends to be too slow and costly for a multi-sensor dynamic modeling system. Debevec et al. [13] use a human editing system with automatic model refinement to recover 3D geometry and a view-dependent texture mapping scheme to texture the model. This 3D recovery method does not map well to our objectives it relies on human editing.

For the purpose of only re-rendering the event, the imagebased rendering approach [14] has been studied extensively. Katayama et al [15] demonstrated that images from a dense set of viewing positions on a plane can be directly used to generate images for arbitrary viewing positions without the need for correspondences. Levoy and Hanrahan [16] and Gortler et al. [17] extend this concept to construct a four-dimensional field representing all light rays passing through a 3D surface. New view generation is posed as computing a 2D cross section of this field. These approaches require a very large number (typically thousands) of real images to model the scene faithfully, making extension to dynamic Scene modeling impractical.

View transform exploits correspondences between images to project pixels in real images into a virtual image plane [18]. View interpolation [19] [20] and view morphing [21] interpolate the correspondences, or flow vectors, to predict intermediate viewpoints. By computing a form of camera calibration directly from the correspondences, these views are guaranteed to be geometrically correct [21]. Virtual

objects can be added into the newly rendered images in a visually consistent manner. However, these solely image- or view-based approaches do not construct an explicit representation of the event, and thus the event itself cannot be manipulated in the computer.

The only large-scale attempt to model dynamic events other than our own is Immersive Video [22]. The stationary parts of the dynamic environment are modeled off-line by hand. The dynamic parts of the event are identified in *each* image by subtracting an image of the background from each input image. The resulting "motion" masks are intersected using *an* algorithm similar to standard shape-from-silhouette methods, resulting in a volumetric model. The final colored model is acquired by back-projecting the input images onto the geometric model. Because shape is recovered from silhouettes, the final model will be unable to identify cavities in the real scene, which places restrictions on the types of objects that can be modeled.

3.4 Virtualized Reality Studio: From Analog "3D Dome" to Digital "3D Room"

Figure 3.2 (a) is a picture of the 3D Dome - our first virtualized reality studio facility. A 5-meter diameter geodesic dome was equipped with 51 cameras placed at nodes and the centers of the **bars** of the dome. They provided viewpoints all around the scene. Color cameras with a 3.6mm lens were used for achieving a wide view (about 90° horizontally). **All** of the cameras were synchronized with a single common sync signal, so that the images taken at the same time instant from different cameras correspond to the same scene. Due to cost, the 3D Dome took the strategy of **real** time recording and off-line digitization. Each of 51 camera outputs **is** recorded by each of 51 consumer-grade VCRs. Every field of each camera's video is time stamped with a common Vertical Interval Time Code (**VITC**). The tapes are digitized individually off-line under the control of a computer program; the computer **can** identify and capture individual fields of video using the time code.

Recently we have upgraded the studio setup to the "3D Room" - a fully digital system that **can** digitize all of the video signal in real time while **an** event occurs [8]. As shown in Figure 3.2 (b), a large number of cameras (at this moment 49 of them) are mounted on the walls and ceiling of the 20 feet × 20 feet × 9 feet room, all of **which are** synchronized with a common **signal**. A PC-cluster computer system (consisting of 17 PCs at this moment) can capture all the video signals from the cameras simultaneously in real time in the 42-2 format **as** uncompressed and unlossy fill frame images with color (640 × 480 × 2 × 30 byte per seconds).

The processing results contained in this paper, however, are all from the data captured by the old 3D Dome.

at man-
ruct an
pulated

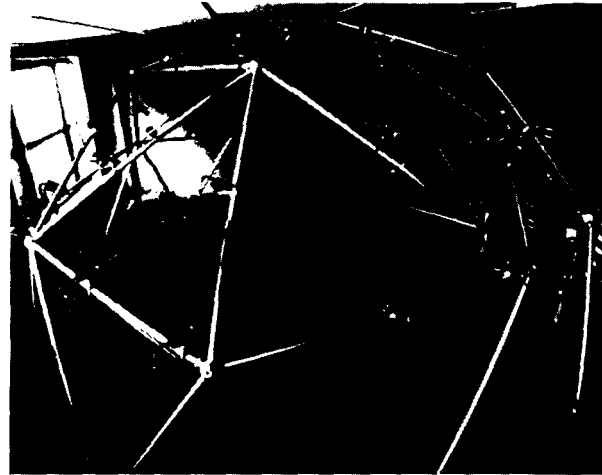
n is Im-
nodeled
by sub-
notion”
houette
ired by
s recov-
;he real

“3D

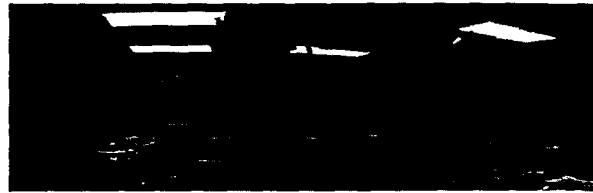
studio
placed
nts all
ving a
with a
t from
ok the
utputs
s video
oes are
mputer

digital
occurs
ent 49
9 feet
mputer
s from
d and

e data



(a)



(b)

Figure 3.2 Studios for Virtualized Reality: (a) 3D dome; (b) 3D room.

3.5 Creation of Three-Dimensional Model

3.5.1 Overview of Processing

Figure 3.3 shows an overview of the current processing method for Virtualized Reality. The input images are passed through an image-based stereo algorithm, generating a range image (a) corresponding to each camera at each time instant. A volumetric integration technique is then applied to recover a single global model by extracting the polygonal approximation to the iso-surface (b) for each time instant. Each model is then reprojected into each camera to create a new range image (c). These range images have much fewer errors than those originally computed by stereo. The stereo process is repeated to obtain the refined range estimate (d) (Section 3.5.5). These are then volumetrically integrated again. Before extracting the final model (f), free space is carved out using foreground-background masks (e).

3.5.2 Stereo Image Matching for Range Image Creation

Stereo algorithms compute depth using triangulation. Strong calibration of the cameras allows computation of the depth in a Euclidean (“world”) coordinate system for

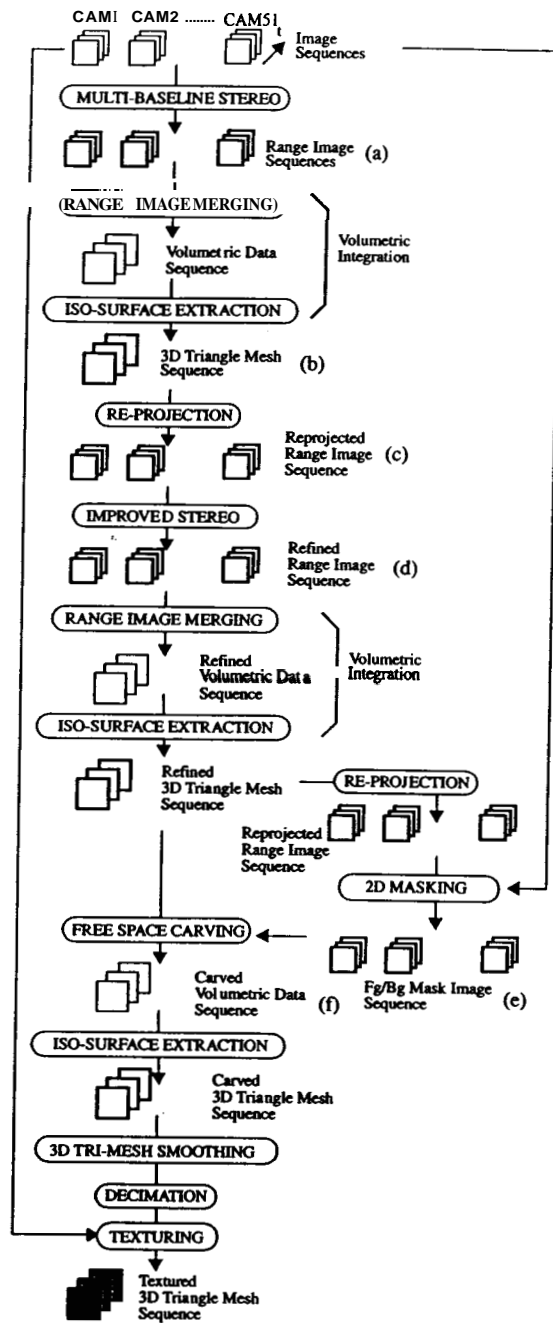


Figure 3.3 Overview of the processing method of creating a Virtualized Reality model.

3.5 Creat

each pixel
baseline
by incorp
marily b
for every
number

Basic s

We begi
that bot
camera
, which
to estim
camera.
second
points,
Ster
the disp
ity bet
ery. Inc
matchin
chances

Adapt

Multi-t
between
are me:
a para
The
depth,
era, in
combin
baselin
date

Robu:
Atmos
compa
other j
as a fu
relatio
does t
each,
relation

non-u

each pixel, given its correspondence(s) in the other image(s). We adapted the multi-baseline stereo algorithm (MBS) [23] for a general, non-parallel camera configuration by incorporating the Tsai camera model [24]. The choice of MBS was motivated primarily by two factors. First, MBS recovers dense depth maps - a depth estimate for every pixel in the intensity image. Second, MBS can take advantage of the large number of cameras to improve the depth estimates.

Basic stereo

We begin by considering a simple stereo system with two parallel cameras. Assume that both cameras are pinhole projectors with the same focal length f . The second camera is laterally shifted down the negative X axis of the first camera by distance b , which is referred to as the baseline between the two cameras. The goal of stereo is to estimate the depth to a scene point corresponding to each pixel in the reference camera. This requires the determination of a corresponding image point in the second camera for each pixel in the reference camera. For any pair of corresponding points, the difference between these image points gives the disparity d :

Stereo searches for the corresponding points along the epipolar line, and selects the disparity yielding the best match. Increasing the baseline amplifies the disparity between corresponding image points, giving better precision for depth recovery. Increasing the baseline, however, also increases the difficulty of finding correct matching points. MBS retains the advantages of a large baseline while reducing the chances of incorrect matches.

Adaptation to MBS

Multi-baseline stereo attempts to improve matching by computing correspondences between multiple pairs of images, each with a different baseline. Since disparities are meaningful only for each pair of images, we rewrite the above equation to derive a parameter that can relate correspondences across multiple image pairs:

The search for correspondences can now be performed with respect to the inverse depth z , which has the same meaning for all image pairs with the same reference camera, independent of disparities and baselines. The resulting correspondence search combines the correct correspondence of narrow baselines with the precision of wider baselines.

Robust correspondence detection

A most typical method to compute correspondences between a pair of images is to compare a small window of pixels from one image to corresponding windows in the other image. The matching process for a pair of images involves shifting this window as a function of x and computing the degree of matching, usually with normalized correlation or sum of squared differences, over the window at each position. The MBS does this computation for all the image pairs, adds the resulting matching scores for each x , and finds the value which shows the best match. We use normalized correlation, which is less immune to the image noises due to viewing angles or camera's non-uniformity.

Figure 3.4 (b) shows an example depth map obtained by MBS for a scene shown in Figure 3.4 (a). The farther points in the depth map appear brighter. We apply the MBS stereo to compute a depth map for each of 51 camera views. In doing so, 3 to 6 neighboring Cameras provide the baselines required for MBS. Because matching becomes increasingly more difficult as the baseline increases, adding more cameras may not necessarily improve the quality of the depth map.

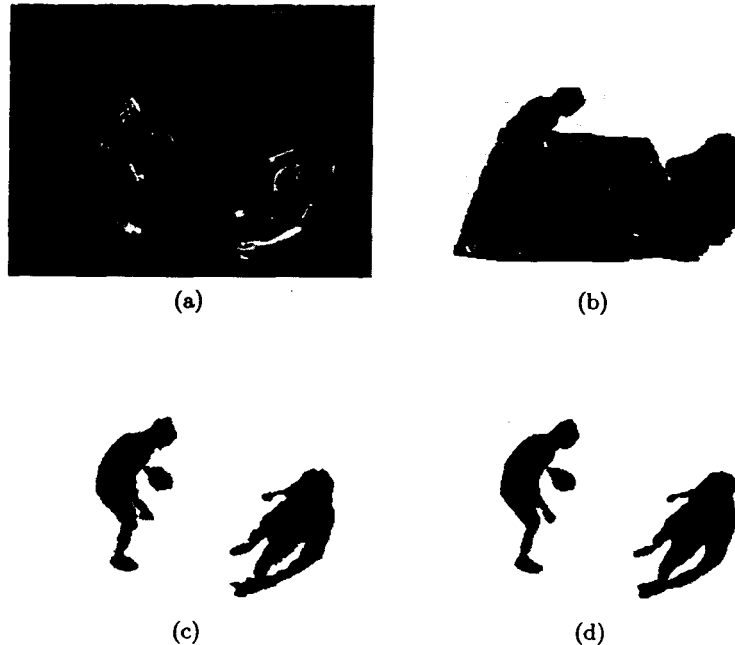


Figure 3.4 (a) Original image (b) Range image from stereo (c) Reprojected range image from first volumetric integration (d) Reprojected from the second volumetric integration.

Any window-based stereo matching has two problems. One is that in image regions with low texture, the depth estimate will have low reliability. A more serious problem is the phenomena of fattening or thinning an object along its boundaries. At the object boundary, a window contains both foreground or background surfaces, for which disparities are different. As a result, the “best” disparity found could be either that of background, foreground, or in-between, depending the strength of their texture patterns. Thus the foreground object becomes either fattened into the background or eaten by the background. Because of these problems, it is unavoidable that the resultant depth map using only stereo includes errors in depth, especially at or near depth discontinuity. Section 3.5.5 will discuss improvements of the MBS stereo results.

3.5.3 Volu

The initial rang each scene. Th model has recei range images [C mal lighting cor problems inher multiple depth tion. We adapt [10], to constr

The scene s mulates the sig image into the by tessellating pixels have a la weight may also estimate reliab tessellated surf map. From thi and negative f the voxel value all range image their values. E ing cubes algo implicit surfac

In order to we made one instead of limiti neighboring vo lies in front of At the same ti distance contr all others in tl the ability of while not sign

3.5.4 Dec

One drawback in the resultin foreground ob The number voxel space, s in the find n decimation al degradation c

3.5.3 Volumetric Merging Depth Maps

The initial range images are then merged, to recover a single volumetric model of each scene. The problem of merging multiple range estimates into a single object model has received much attention recently, but mainly working with high quality range images [9] [10] [25] [26]. The depth maps generated by stereo under normal lighting conditions (i.e., no structured lighting or special textures) suffer from problems inherent in window-based correlation, as described above. Fusion of the multiple depth maps of the same scene can improve the precision of surface localization. We adapted a volumetric integration method, proposed by Curless and Levoy [10], to construct a global 3D surface model of the scene [6].

The scene space to be modeled is divided into small voxels. Each voxel accumulates the signed distance to the surfaces in the range images. To add a range image into the volume, the image is first converted into a set of triangular surfaces by tessellating the image, connecting each pixel to its neighbors. If neighboring pixels have a large difference in depth, no tessellation occurs between the pixels. A weight may also be attached to each range estimate, allowing incorporation of range estimate reliability into the fusion process. Next, each voxel is projected onto the tessellated surface along the line of sight of the sensor providing the current depth map. From this projection, the signed distance from the surface (positive for front and negative for rear, for example) to the voxel is computed, and it is added to the voxel value. The process is repeated for each voxel. After accumulating across all range images, the voxels implicitly represent the surface by the zero crossings of their values. Extraction of an implicit surface, or iso-surface, is done by the marching cubes algorithm [27] [28], which generates 3D triangle meshes representing the implicit surfaces.

In order to overcome the false surfaces (fattening or thinning) generated by stereo, we made one noteworthy change to the original Curless and Levoy algorithm. Instead of limiting the extent of contributions by each tessellated surface only to the neighboring voxels, we allowed the algorithm to adjust the values of all voxels which lies in front of the surface as viewed from the sensor that has generated this surface. At the same time, for voxels far in front of the surface, we clamp the weighted, signed distance contribution of each viewpoint so that this single view does not overwhelm all others in the fusion process. This modification gives significant improvement in the ability of the algorithm to reject the numerous outliers in our range images, while not significantly degrading the recovered shape [6].

3.5.4 Decimation and Texturing

One drawback of the volumetric merging algorithm is the large number of triangles in the resulting models. For example, fusing range images of our dome itself, with no foreground objects, at the 1-cm resolution, created a model with 1,000,000 triangles. The number of triangles in the model is directly related to the resolution of the voxel space, so increasing the voxel resolution will increase the number of triangles in the final model. To reduce the number of triangles, we apply an edge-collapse decimation algorithm [29]. Typically, we obtain a reduction of 20:1 without visible degradation of mesh quality.

3.5.5 Stereo Improvement and Foreground/Background Separation

By projecting the triangle mesh obtained by depth map merging into the depth buffers of the original cameras, we obtain an approximate range image, with values for depth at all the foreground pixels. Using this approximate value as much tighter bounds on the range of depth values for the pixel, we perform a second round of stereo matching. This process can significantly improve the stereo results. The improved range images from this iteration are again merged to yield a more accurate 3-D model.

Figure 3.4 shows the results by this refinement process. Compared with the initial range image (Figure 3.4 (b)), one can observe progressive improvements in the reprojected range image from volumetric integration (Figure 3.4 (c)) and the final model obtained from the second round of merging.

The volumetric integration process creates a 3D triangle mesh representing the surface geometry in the scene. To complete the surface model, a texture map is constructed by projecting each intensity (or color) image onto the model and accumulating the results. A simple approach would be to average the intensity from all images in which a given surface triangle is visible, but a more sophisticated method is used to learn a view-dependent texture map. At the time of re-rendering, the contributions of multiple views are weighed so that the most "direct" views dominate the texture computation, and the multiple views are super-resolved for improving the image quality [1].

Figure 3.5 shows a comparison of image quality between an original image (a) and the images rendered from the constructed model (b)-(d). Figure 3.6 (b) shows a rendered textured image from a position close to the original camera. Figures (c) and (d) show rendered images from virtual viewpoints far away from the real camera. We see that good shape and texture recovery ensure that the quality of the rendered images closely match that on the original image, both when the virtual viewpoint is near an original image, and when it is away from it. Note that the original images have been digitized only with a half resolution of NTSC (512 x 256) in this example.

3.6 Combining Multiple Events

There is often a desire to combine many different kinds of events that occurred separately in space and time. Also, a large environment may be created by modeling its individual components separately. Virtualized Reality allows such spatio-temporal integration of event models that are created separately.

Each virtualized reality model is typically represented by a triangle mesh, with a list of vertex coordinates, texture coordinates, and polygon connectivities. The vertex coordinates of each such model are defined independently with respect to a local coordinate system. To combine different models, a simple rotation and translation is applied to the vertex coordinates alone of each triangle mesh, so that each local origin is mapped to the location of the world origin with the desired orientation.

A little more attention needs to be paid to integrating the temporal components.

A

Figure 3.5 (a) Original point; (c) and (d) Virtu:

Firstly, if one or more virtual sequence, those Or in some cases, the e Once this is done, each frame on the global ti

In addition, since can be also integrated CAD programs.

3.7 Example

"Basketball One-on-O Figure 3.6 shows an e the event, two people ure 3.6 (a) shows two time-varying 3D mode presentation) clearly : hair of the woman and

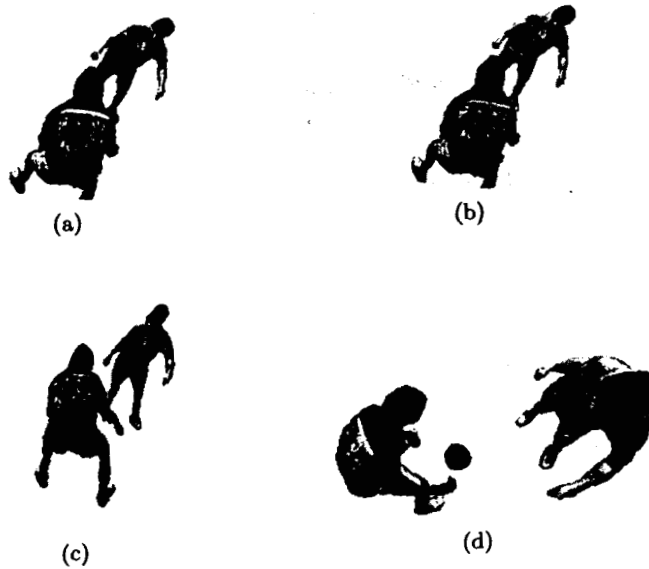


Figure 3.5 (a) Original image (masked); (b) Virtual image rendered from the same view point; (c) and (d) Virtual images from other view points.

Firstly, if one or more sequences are not modeled at the frame rate of the desired virtual sequence, those sequences would need to be subsampled or supersampled. Or in some cases, the event is used in a reverse order, as in the example shown later. Once this is done, each time frame on component event sequence is mapped to a frame on the global time scale.

In addition, since our models are a metric description of the real event, they can be also integrated with traditional VR models, such as those often produced by CAD programs.

3.7 Examples

"Basketball One-on-One"

Figure 3.6 shows an example where a single 10-second event was digitized. During the event, two people move and a ball is passed from one person to the other. Figure 3.6 (a) shows two frames of the input image sequence. Figure (b) shows the time-varying 3D model of the event; the video sequence of this model (shown at the presentation) clearly shows that detailed motions, such as motion of the pony tail hair of the woman and flapping of man's suit, are correctly and faithfully digitized.

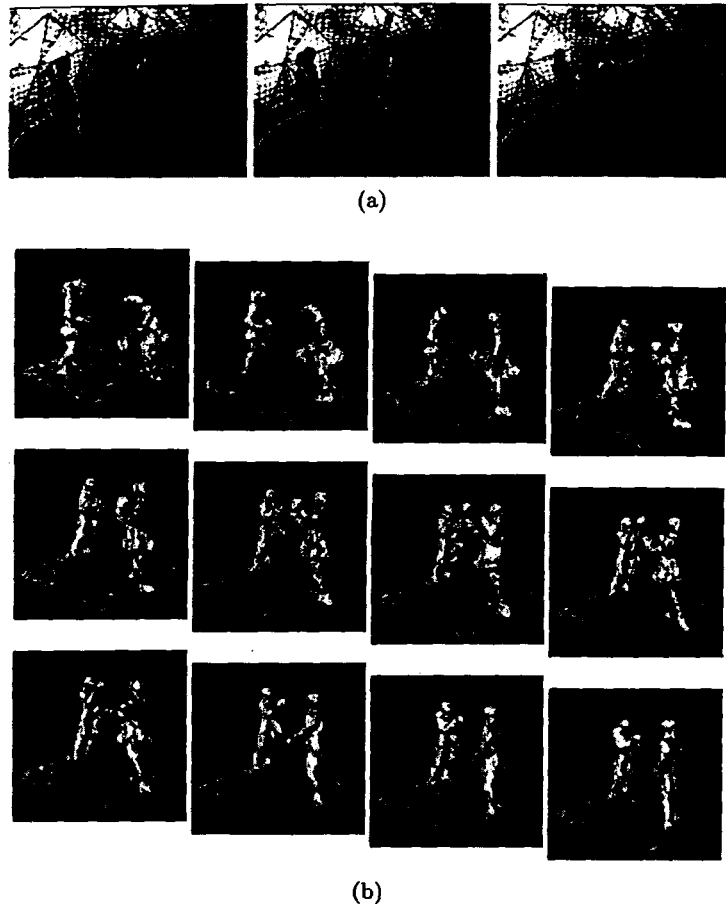


Figure 3.6 The Virtualized Reality model of "basketball one-on-one" event. (a) input images; (b) a time-varying 3D model.

"Three-man Basketball"

In our example, two separate events are recorded. The first event involves two players, where one player bounces a basketball and passes it off to the side while the other attempts to block the pass. The second event involves a single player who receives a basketball and dribbles the ball. Since we are free to choose frames in reverse, we actually record the second event by having the player throw the ball to the side. Both these events are recorded separately, so no camera ever sees all three players at once. The volumetric models for the first time frame for each of these models are shown in Figures 3.7 (a) and (b), respectively. The aim is to combine the events, so that the final model contains a motion sequence where the first player passes the ball to the third player, as the second player attempts to block this pass.

Figure 3.7
second eve

The sp
the first e
construct
second eve
is physica
of the fir
the third.

Figure
spatial an
Figure 3.9
how the s
The vi
basketball
as the car
original ir
easy for tl



Figure 3.8 The combined volumetric model of an instant.



Figure 3.9 Time lapse volumetric model of the entire sequence of the combined event. (a) and (b) display the model from two angles.

3.8 Conclusions

Our Virtualized Reality system provides a significant new capability in creating virtual models of time-varying large-scale events involving free-form and large objects such as humans, using multiple video streams. In addition to the modeling, we have the capability to produce synthetic video from a varying virtual viewpoint. There are no restrictions on the positions from which virtual views can be synthesized. The system is complete to go from captured real image sequences to virtual image sequence, with no human input required with regard to knowledge or structure of the scene. In addition, it provides the capability to integrate two or more independent motion sequences with each other, or with an existing static or dynamic VR model.

The model creation process described in this paper is a two-stage process: stereo matching for 2-1/2 D surface extraction from each view and merging them into a single representation. After all, the quality of the results is strictly dependent on

References



1



5

Figure 3.10 Camera trajectory moving in-between

the quality of the 3D model directly similar to voxel allows us to explore the image rendering

Reference

- [1] T. Kanade and early Boston, pp
- [2] T. Kanade mobile in and Conf.
- [3] T. Kanade for video-CVPR '96.
- [4] P. J. Narasimhan every CMU-RI-
- [5] P. J. Narasimhan using dense pp.3-10, J

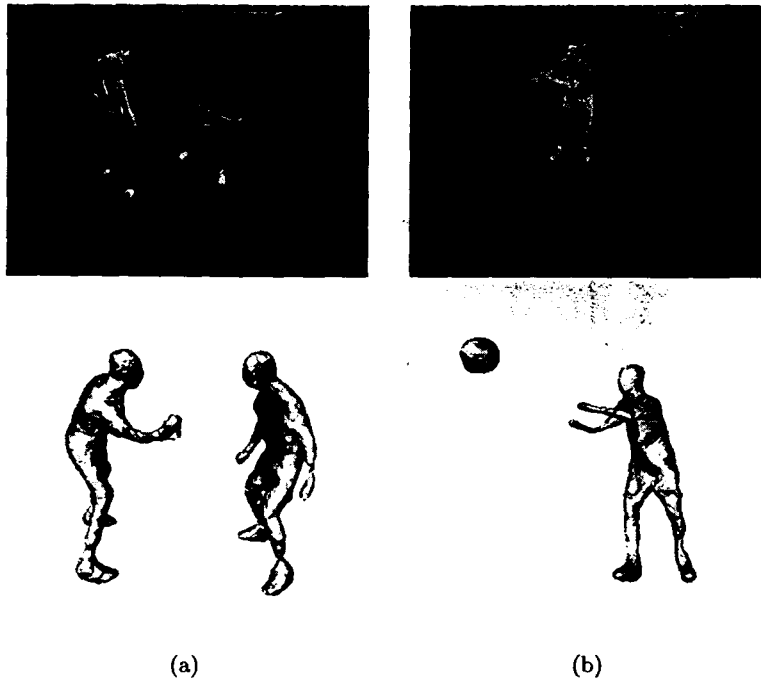


Figure 3.7 The volumetric model: (a) one of the frame instant of the first event (b) the second event (see color pages).

The spatial transform is done so that the ball at the end of the last frame of the first event coincides with the ball at the beginning of the second event. The constructed models for the last frame of the first event and the first frame of the second event are modified to remove the ball (this is a simple operation since the ball is physically disjoint from the rest of the scene) and this is used for time instances of the first event after the ball has left the scene, i.e. while it is being dribbled by the third player.

Figure 3.8 shows the volumetric model obtained by combining both events. The spatial and temporal smoothness of the volumetric models generated can be seen in Figure 3.9, which shows a time-lapse volumetric model over all time frames, that is how the space is occupied by the combined event.

The virtualized reality model of the combined event is placed inside a virtual basketball court. Figure 3.10 shows a several images of a flythrough of the event, as the camera flies over and in-between the players - the non-existent views in the original input sequence. In this case, the virtual model is static in time, but it is easy for the virtualized model to be integrated with a time-varying virtual model.

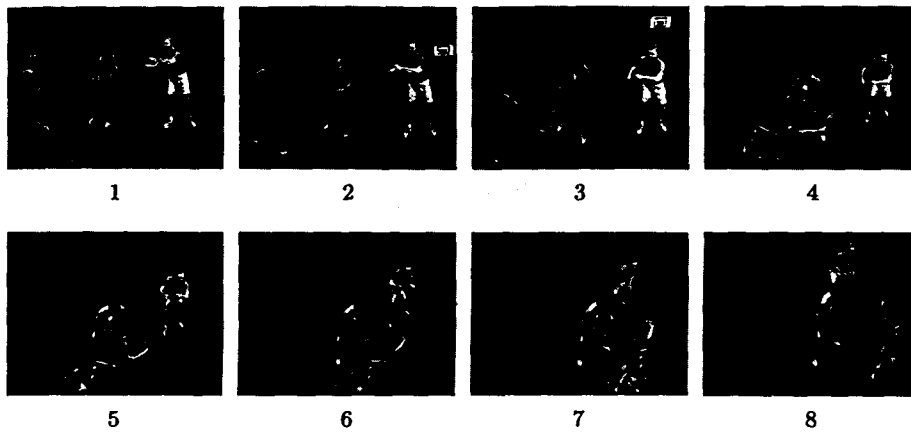


Figure 3.10 Images of a fly-by of the combined event on a basketball court. The virtual camera trajectory includes a spiral motion going up and down around the players and moving in-between the players (see color pages).

the quality of correspondence. Currently we are working on a method to create a 3D model directly into a voxel representation by fusing multiple images in a manner similar to voxel coloring by Seitz and Dyer [30]. Also, it appears that this methods allows us to exploit multi-image information more systematically, as well as allowing the image rendering process into the voxel representation.

References

- [1] T. Kanade, P. J. Narayanan, and P. W. Rander: "Virtualized Reality: Concepts and early results," *IEEE Workshop on the Representation of Visual Scenes*, Boston, pp.69-76, Jun. 1995.
- [2] T. Kanade, P. J. Narayanan, and P. W. Rander: "Virtualized Reality: Being mobile in a visual scene," *Int'l Conf. on Artificial Reality and Tele-Existence and Conf. on Virtual Reality Software and Technology*, Japan, Nov. 1995.
- [3] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka: "A stereo machine for video-rate dense depth mapping and its new applications," *Proc. IEEE CVPR'96*, San Francisco, CA, Jun. 1996.
- [4] P. J. Narayanan, P. W. Rander, and T. Kanade: "Synchronizing and capturing every frame from multiple cameras," Robotics Institute Technical Report, CMU-RI-TR-95-25, Carnegie Mellon Univ., 1995.
- [5] P. J. Narayanan, P. W. Rander, and T. Kanade: "Constructing virtual worlds using dense stereo," *Proc. Sixth Int'l Conf. on Computer Vision*, Bombay, India, pp.3-10, Jan. 1998.

nd event.

creating
objects
we have
. There
hesized.
l image
e of the
pendent
model.
: stereo
t into a
lent on

- [6] P. W. Rander, P. J. Narayanan, and T. Kanade: "Recovery of dynamic scene structure from multiple image sequences," *Int'l Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Washington, D.C., pp.305-312, Dec. 1996.
- [7] P. W. Rander, P. J. Narayanan, and T. Kanade: "Virtualized Reality: An immersive visual medium," *IEEE Multimedia*, vol.4, no.1, pp.3447, Jun. 1997.
- [8] T. Kanade, H. Saito, and S. Vedula: "The 3D room: Digitizing time-varying 3D events by synchronized multiple video streams," Technical Report, CMU-RI-TR-98-34, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, Dec. 1998.
- [9] A. Hilton, J. Stoddart, J. Illingworth, and T. Windeatt: "Reliable surface reconstruction from multiple range images," *Proc. ECCV'96*, pp.117-126, Apr. 1996.
- [10] B. Curless and M. Levoy: "A volumetric method for building complex models from range images," *P m . SIGGRAPH'96*, Aug. 1996.
- [11] M. D. Wheeler, Y. Sato, and K. Ikeuchi: "Consensus surfaces for modeling 3D objects from multiple range images," *Proc. DARPA Image Understanding Workshop*, 1997.
- [12] Y. Sato, M. D. Wheeler, and K. Ikeuchi: "Object shape and reflectance modeling from observation," *P m . SIGGRAPH'97*, pp.379-388, 1997.
- [13] P. Debevec, C. Taylor, and J. Malik: "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," *Proc. SIGGRAPH'96*, Aug. 1996.
- [14] L. McMillan and G. Bishop: "Plenoptic modeling: An imagebased rendering system," *P m . SIGGRAPH'95*, Los Angeles, 1995.
- [15] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura: "A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images," *P m . SPIE: Stereoscopic Displays and Virtual Reality Systems II*, vol.2409, pp.11-20, 1995.
- [16] M. Levoy and P. Hanrahan: "Light field rendering," *P m . SIGGRAPH'96*, Aug. 1996.
- [17] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen: "The Lumigraph," *P m . SIGGRAPH'96*, Aug. 1996.
- [18] S. Laveau and O. Faugeras: "3-D scene representation as a collection of images," *P m . ICPR'94*, 1994.
- [19] E. Chen and L. Williams: "View interpolation for image synthesis," *P m . SIGGRAPH'93*, 1993.
- [20] S. Vedula, P. W. Rander, H. Saito, and T. Kanade: "Modeling, combining, and rendering dynamic real-world events from image sequences," *Proc. 4th Conf. on Virtual System and Multimedia*, vol.1, pp.326-332, Gifu Japan, Nov. 1998.

- [21] S. M. Seitz, 1996.
- [22] R. Jain and M. S. Kuhlman, *Conf. on*
- [23] M. Okutani, *Pattern*
- [24] R. Tsai, *chine vi Robotics*
- [25] H. Hopmann and J. Schwinger, *Proc. S*
- [26] G. Turk, *SIGGR*
- [27] J. Blostein, *ed.) G edu/pu*
- [28] W. Lorincz, *constru*
- [29] A. Johnson, *Institu*
- [30] S. M. Seitz, *oring,* 1997.
- [31] S. B. Gortler, *multib*
- [32] T. Weiskopf, *view i*

- nic scene
r Fusion
12, Dec.
- lity: An
in. 1997.
- varying
, CMU-
c. 1998.
- face re-
'6, Apr.
- models
- odeling
tanding
- model-
- ecture
n. SIG-
- dering
- ndent
SPIE:
1995.
- , Aug.
- aph,"
- ges,"
- SIG-
- and
f. on
- [21] S. M. Seitz and C. R. Dyer: "View morphing," *Proc. SIGGRAPH'96*, pp.21-30, 1996.
- [22] R. Jain and K. Wakimoto: "Multiple perspective interactive video," *Proc. IEEE Conf. on Multimedia Systems*, May 1995.
- [23] M. Okutomi and T. Kanade: "A multiple-baseline stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.15, no.4, pp.353-363, 1993.
- [24] R. Tsai: "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE J. Robotics and Automation*, vol.RA-3, no.4, pp.323-344, 1987.
- [25] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle: "Piecewise smooth surface reconstruction," *Proc. SIGGRAPH'94*, 1994.
- [26] G. Turk and M. Levoy: "Zippered polygon meshes from range images," *Proc. SIGGRAPH'94*, Jul. 1994.
- [27] J. Bloomenthal: "An implicit surface polygonizer," in (P. Heckbert, ed.) *Graphics Gems IV*, pp.324-349, 1994. (<ftp://ftp-graphics.stanford.edu/pub/Graphics/GraphicsGems/GemsIV/GGemsIV.tar.Z>).
- [28] W. Lorensen and H. Cline: "Marching cubes: A high resolution 3D surface construction algorithm," *Proc. SIGGRAPH'87*, pp.163-170, Jul. 1987.
- [29] A. Johnson: "Control of mesh resolution for 3D computer vision," Robotics Institute Technical Report, CMU-RI-TR-96-20, Carnegie Mellon Univ., 1996.
- [30] S. M. Seitz and C. R. Dyer: "Photorealistic scene reconstruction by voxel coloring," *Proc. Computer Vision and Pattern Recognition Conf.*, pp.1067-1073, 1997.
- [31] S. B. Kang and R. Szeliski: "3-D scene data recovery using omnidirectional multibaseline stereo," *Proc. IEEE CVPR'96*, San Francisco, CA, Jun. 1996.
- [32] T. Werner, R. D. Hersch, and V. Hlavac: "Rendering real-world objects using view interpolation," *IEEE Int'l Conf. on Computer Vision*, Boston, 1995.