Virtualized Reality: Perspectives on 4D Digitization of Dynamic Events

Takeo Kanade
Carnegie Mellon University

P.J. Narayanan
International Institute of Information Technology

Digitally recording dynamic events, such as sporting events, for experiencing in a spatio-temporally distant and arbitrary setting requires 4D capture: three dimensions for their geometry and appearance over the fourth dimension of time. Today's computer vision techniques make 4D capture possible. The Virtualized Reality system serves an example in this discussion on the general problem of digitizing dynamic events.

Dynamic events spark great interest for entertainment, education, and exploration. In traditional media, each viewer observes the event from a fixed viewpoint determined by the producer or the viewer's seating position. Can we digitally record a dynamic event and experience it in a spatio-temporally flexible manner? A viewer can watch a sporting event from any viewpoint—for example, a player's or even the ball's. An audience member can view a ballet from a virtual seat in the middle of the stage. The recording can enhance the training experience so that medical personnel can visit a particularly difficult step repeatedly from the most informative viewpoint. We can also edit or modify the recorded event in creative ways, such as changing its appearance after capture or creating a new event by combining parts of recorded events. Such a medium of capturing and manipulating digitized dynamic events can open new vistas in immersive and participative entertainment, empowering viewers to control their experience.

Telepresence is the ability to be present at an event taking place at a distance. If we can digitize a dynamic event and let a user immerse into it, we can achieve tele-experience (if live) or post-experience (if delayed), navigating through and interacting with the digitized event. The ability to experience a remote dynamic event in its richness, unhindered, arguably can provide the functional equivalent of teleportation.

In the early 1990s, we began developing multicamera computer-vision technologies at Carnegie Mellon to digitize large, dynamic events. The CMU system comprised a large number of cameras to capture an event inside a room from all directions. It produced the 4D event description, consisting of the 3D model of the scene and its appearance across time. We used tools similar to those that VR uses to render the digitized event, either in real time or at a later time. We coined the term *Virtualized Reality* to emphasize the aspect of converting real events to virtual ones. In this article, we present the Virtualized Reality system's details from a historical perspective.

Origins of the field

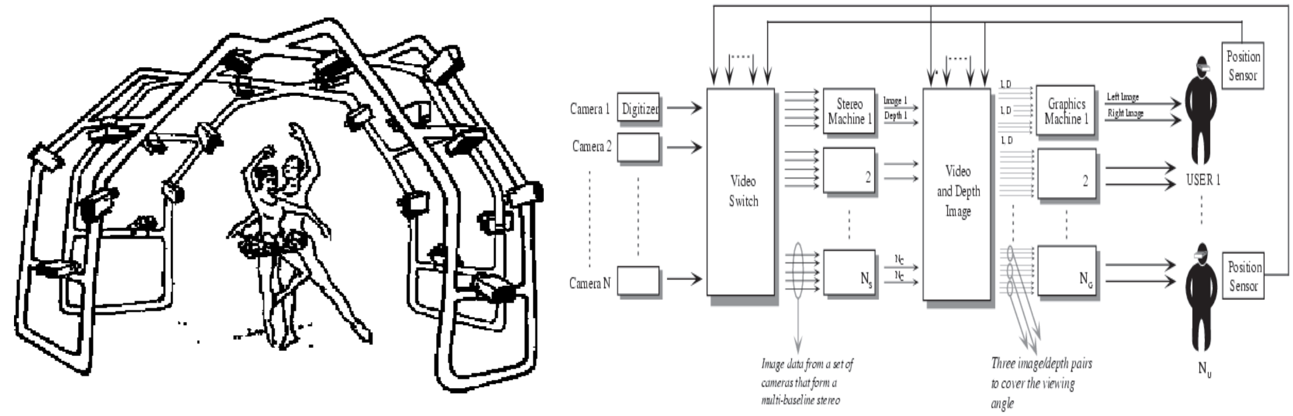
The precursor to the Virtualized Reality project was the development of the multicamera, multibaseline video-rate stereo machine in the early 1990s. By 1993, we built a series of machines that could convert the input scene to a 256 × 256, 8-bit depth map at a speed of 30 frames per second.[1] With the machine, we demonstrated a new real-time image-merging technique, named z-key. Like the blue key, the z-key switches pixel-by-pixel a real scene and a virtual scene but by using distance instead of color as the key.[2]

Having realized that we could digitize a dynamic scene as a whole if multiple cameras observed it from multiple directions, we built in mid-1994 the first Virtualized Reality system with 10 cameras, and then expanded it in late 1995 to a 51-camera dome system that could capture an event from a complete hemisphere. These earlier systems were analog and offline; videos were synchronized and recorded on video tapes with time codes and digitized later for processing. An upgrade to the system came in 1998 with a 49-camera digital room, and again in 2002 with the current facility of a 48-camera large space, where all of the capturing is done completely digitally and online.

Obtaining an object's or small space's geometric structure and showing it as a textured model was standard practice in computer vision. The Virtualized Reality system, however, was one of the first systems to capture a large time-varying dynamic event with a significant number of cameras and to turn the recording into a space-time representation with the intent of experiencing it later.[3,4] Several efforts with similar goals appeared

Published by the IEEE Computer Society

0272-1716/07/$25.00 © 2007 IEEE



1 **Conceptual block diagram of the Virtualized Reality system.**

thereafter. The Multiple Perspective Interactive Video project used a combination of static models, change detection, and shape from triangulation to model and navigate through large spaces.[5] Later, Image-Based Rendering (IBR) techniques for capturing objects from multiple viewpoints and generating novel viewpoints[6-8] became a topic of intensive study, initially for individual static objects, and later for dynamic scenes. National Research Council's 3D modeling project used various modeling techniques for buildings, heritage sites, and mines.[9] The Digital Michelangelo project[10] and the Great Buddha project[11] used high-quality range finders for archiving and preserving cultural heritage. More recently, a revival of interest has grown around capturing dynamic events, including the blue-c system,[12] the 3D Video Recorder from ETH Swiss Federal Institute of Technology,[13] the Free-Viewpoint Video system from the Max Planck Institute for Computer Science,[14] and the video-based rendering system from Microsoft Research.[15] Nagoya University's system for free-viewpoint TV to capture dynamic events has several thousand cameras arranged in an array[16] and generates new views with ray-space interpolation methods.

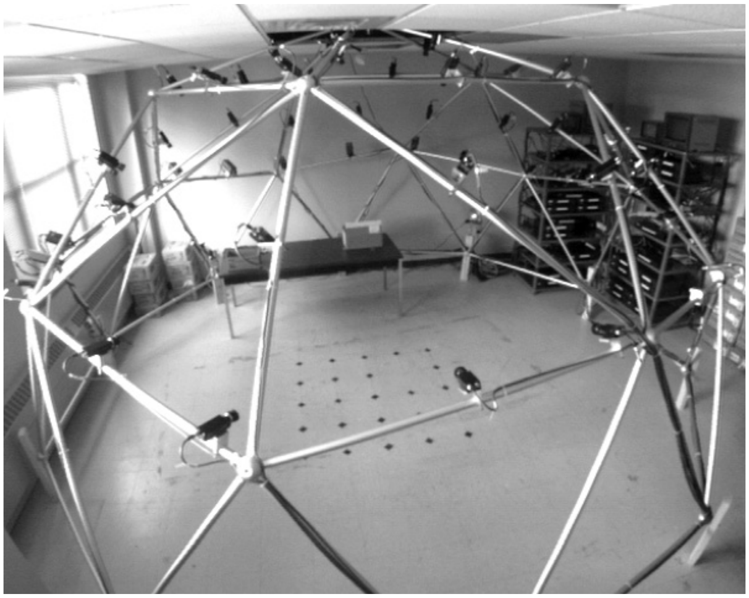
The Carnegie Mellon Virtualized Reality system, designed to capture a large-scale dynamic event, evolved over the years from being simple and small to being sophisticated and large.

Virtualized Reality system for dynamic event digitization

Figure 1 shows the early conceptual block diagram of the Virtualized Reality system.[4] The process begins with the nonintrusive capture of the dynamic events.

Nonintrusive capture

The system captures a dynamic event by using multiple cameras at different points in the event space. Cameras are ideal capturing devices because of their nonintrusiveness, speed, economy, familiarity, and universality. Multiple cameras provide the complete surrounding view of an event and facilitate structure recovery using structure from motion techniques. Cameras remain the more economical solution, though embedding imperceptible patterns and infrared light have also been tried.



2 **The first Virtualized Reality setup (circa 1995) with the dome, cameras, and VCRs.**

System setup. The first significant Virtualized Reality setup built in 1995, called the 3D Dome, used 51 cameras mounted on a geodesic dome, 5 m in diameter (see Figure 2). It used industrial grade National Television Standards Committee (NTSC), monochrome (later replaced by color), analog charged coupled device (CCD) video cameras for image capture. The system used lenses with 3.6 mm focal length for a field of view close to 90 degrees. The cameras' arrangement provided all-around views of the event, and they were close enough to each other so that the computer-vision stereo algorithms worked well. The cameras looked at the center of the dome and had a volume of intersection close to 3 m × 3 m × 2 m. The cameras were synchronized with a common sync signal, and the Vertical Time Code Interval (VITC) was inserted into their video output. The system recorded output of each camera on a separate consumer-grade S-VHS VCR for later digitization and processing. The cost of the setup was about US$1,000 per channel. Direct digital capture of multiple video channels at this scale was not realistic at that time. (For details on this capture system, see the technical report.)17

3 (a) The 3D Room setup with digital capture (circa 1998). (b) A panoramic view of the current Virtualizing Studio (circa 2002) with 48 cameras and full digital capture. (c) The outside view of the Virtualizing Studio.

Direct digital capture can provide higher-quality images than using analog tapes. The image quality has impact on the captured event's textural appearance as well as on its structure computed from them. We built the 3D Room facility, shown in Figure 3a, in 1998. The 3D Room system used 49 cameras and captured the images directly using 17 PCs. The memory size limited the capture to about 800 frames per PC, or just under 18 seconds of event time at 15 frames per second. The cost of the setup was still about $1,000 per channel, with the extra cost of color cameras offset by that of the VCRs.

Direct capture to a secondary storage device became feasible as the technology advanced. The Virtualizing Studio system, built in 2002, can capture the output of all cameras directly onto the hard disks. This facility uses 48 color cameras for event capture and has increased event space to 6.1 m × 6.7 m × 4.3 m. Nine cameras are mounted on the ceiling and the rest along the walls at two heights. The system uses high-end 3CCD cameras with automatic zoom. Figure 3b shows the panoramic view of two side walls of the studio, and Figure 3c the outside view. Studio-quality lighting in the room alleviated some of the problems we experienced in earlier setups. Hours of recording using 48 cameras in full color is possible with this setup, with the only limit being the available disk space. The cost per channel is higher at approximately $8,000 for this setup due to using the 3CCD cameras.

Because the technologies of cameras, buses, memory, and disk storage have advanced much further, it's possible today to digitally capture the outputs of numerous cameras and stream them directly to the disk. IEEE 1394 or Firewire cameras of resolutions 1,024 × 768 and beyond are available today. Firewire also gives sufficient bandwidth to capture and store three to four cameras to the disk on a single PC. MPEG-2 Transport Stream camcorders providing HDTV resolutions are now appearing in consumer and professional ranges. Recently built 4D-digitization systems take advantage of these new developments. The MPI system uses seven Firewire cameras for capture. The ETH system uses three 3D bricks, each with three Firewire cameras, for capture. The MSR system uses eight Firewire cameras arranged roughly along a line and captures the action live.

Frame synchronization and labeling. The multicameras must sample the same dynamic event in discrete and synchronized time intervals. Synchronized multicamera capture requires two steps. First, the system must synchronize the camera frames to one another. Supplying a reference video signal as the genlock to all cameras will keep them in sync and ensure that they sample the world simultaneously. The second step is to line up the frames or time instants from different cameras. This requires giving a unique label to each frame of all cameras. The Virtualized Reality system achieves this using the Society of Motion Picture and Television Engineers standard VITC mechanism for frame labeling.17

Firewire has emerged as a standard for connecting digital cameras. The Firewire bus has signals that help in synchronizing up to four cameras to the frame level. Commercial devices are available that can synchronize multiple Firewire buses for scaling beyond four cameras. The MSR system uses Firewire-based synchronization. The MPI and ETH systems use externally triggered Firewire cameras.

Camera calibration. The cameras must be calibrated to a common coordinate system if their outputs must correlate with one another. Calibration is critically important, as errors in calibration can distort 3D reconstructions systematically and amplify errors in subsequent processes. Calibrating numerous cameras to a common reference frame is challenging, especially if the cameras are arranged in the outside-in configuration—that is, to cover a space from all sides. The system's scalability requirements imply that the calibration procedure should be simple and extensible to a large number of cameras.

Strong calibration algorithms are well researched and stable implementations exist. A strongly calibrated setup allows full 3D Euclidean reconstruction and facilitates handling the recovered models using standard tools. The Virtualized Reality system used the calibration scheme by Tsai[18] or Zhang.[19] Recent efforts from ETH, MPI, and MSR used a variation on the same schemes. Calibrating a large space requires objects, often specially constructed, with known dimensions that are visible to all cameras. For a truly large space, the system can perform calibration for nearby camera groups, with significant overlap of fields of view in cameras between adjacent groups. However, users should ensure that calibration errors do not accumulate when using such a procedure.

Event modeling and representation

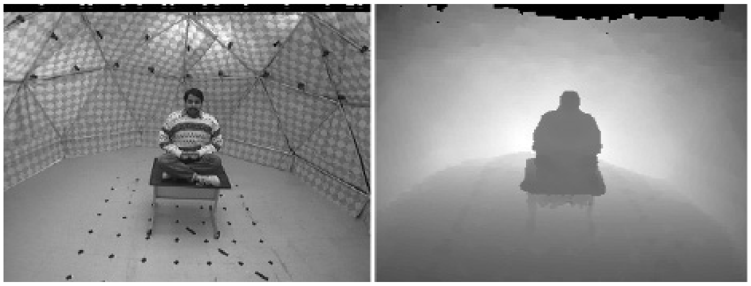
Experiencing the event, either live or later, requires the ability to immerse a viewer in it by placing a virtual camera anywhere in the event space. If hundreds of real views of the event are available, we can generate a pseudo-immersive experience by switching to the view closest to the one the viewer demands. The EyeVision system, an off-shoot of the Virtualized Reality project that debuted at the 2001 Super Bowl football championship, created such an illusion of immersion. It involved coordinated tracking and zooming of about 30 cameras placed on the second deck of the football stadium roughly in a circular arrangement, and, by sequencing through their views for each time instant, the system created a movie *Matrix*-like outside-in spinning replay. Many mosaicing techniques, on the other hand, provide inside-out spinning of a scene by stitching together many images taken from a central viewpoint. Such techniques cannot provide unrestricted immersion as they cannot generate views due to the viewer's own translational motion or inward views from outside. For a complete and unrestricted immersion, a suitable 3D model of the event, explicit or implicit, is necessary.

Scene structure

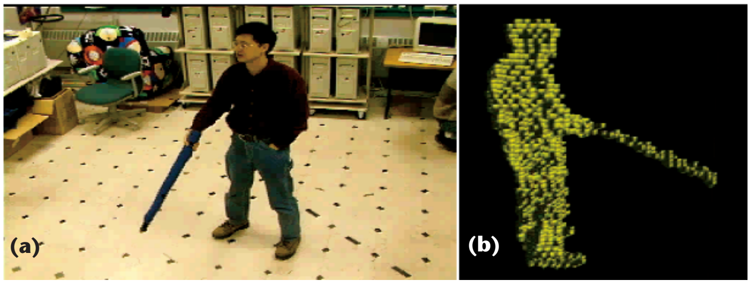
The Virtualized Reality system uses a multicamera stereo algorithm to recover the scene structure. A dense-stereo program takes images from a combination of cameras and computes the depth for each pixel in the image.[1] Figure 4 shows a camera image of a scene and the reconstructed depth map corresponding to it. Such a 3D structure is view-dependent and is computed with respect to each of the cameras. While we can produce immersive display using multiple view-dependent structures, we can also create a global model of the scene in the form of a textured, triangulated model.[4] We do so by merging the individual depth-maps in a volumetric space using a volumetric merging algorithm. Figures 6 and 9 include examples of such constructed complete surface models (CSMs).

The MSR system uses a stereo algorithm based on color segmentation to compute the depth maps with respect to each of the eight cameras used for acquisition.[15] Disparity smoothing removes noise's effects. The ETH system uses three 3D video bricks, each of which consists of three cameras and a projector for active lighting. In alternating frames, the system captures images with and without light projection to obtain accurate structure and unmodified texture, respectively. A representation of the scene as a cloud of points can be rendered using standard point-based rendering techniques.

The shape from silhouette (SFS) technique computes an object shape's visual hull by intersecting the cones corresponding to the object's silhouettes from many views. We can use the SFS in place of stereo or to enhance the stereo algorithms, especially if models of individual objects or persons are being sought. The Virtualized Reality system also uses the SFS for real-time voxel reconstruction of human motion, shown in Figure 5, and for detailed human kinematic modeling.[20] The MPI system also computes a global model as the intersection of silhouettes combined with photo-consistency enforcement for reducing outliers' effects.



4 Reference images and the dense depth map computed using multibaseline stereo.



5 Real-time 3D-voxel reconstruction of a dynamic scene. (a) An input image and (b) a 64 × 64 × 64 voxel representation for 2 m × 2 m × 2 m space computed at 15 frames per second.

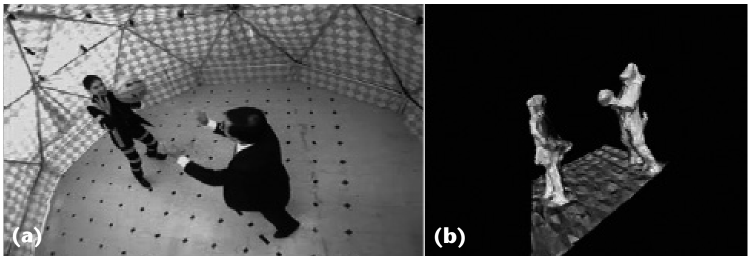
Rendering events from new arbitrary viewpoints

Providing visual experience of a digitized dynamic event requires rendering the scene from an arbitrary viewpoint. An observer has the impression of walking or flying through the event independently if shown a succession of views along his or her path of motion.

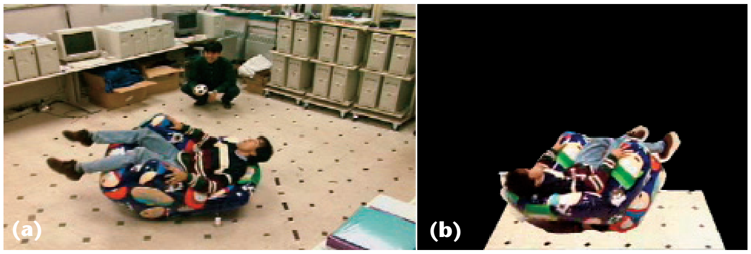
Using the depth maps. We can convert the depth map to a 3D triangulated surface model by constructing two triangles for each 2 × 2 section using a diagonal. If the difference of depth values along any side of a triangle exceeds a certain threshold, that instance is determined as an occlusion boundary and the surface model is deemed to have a hole there. This scheme converts the structure computed by stereo to a visible surface model (VSM). To generate new views, the system can render a VSM with the texture taken from the camera images.

It is also possible to generate new views using a combination of VSMs but without creating a single global model like CSM. Given a desired view's orientation, we can choose the VSM whose original viewing direction is closest to it as the reference VSM to generate most of the desired view. When the system identifies hole regions in the reference VSM, it renders neighboring VSMs to fill the holes and blend with nearby parts. This approach combines view-dependent geometry and texture for view-generation by using the closest VSMs. (See Narayanan et al. for more details on view generation using this method.)[4]

The MSR system[15] renders exactly two VSM models like the scheme we mentioned, but uses matting near occlusion boundaries. By interpolating the colors near the boundary, their scheme is able to produce a continuous appearance, even when the appearance differs between the VSMs.



6 (a) An input frame from a two-person game sequence. (b) The CSM surface model constructed shown without texture.



7 Appearance-based virtual view generation from multicamera videos captured in the 3D room. (a) An input view, and (b) a novel view.

8 Inserting a virtual object in the virtualized scene and making it interact with the real object. The lamp's shadow is cast on the real person consistently with its position and surface shape.

Using the CSM. We can render the CSM using standard graphics algorithms. Real-time navigation of the CSM requires high rendering capabilities, because the model constructed using volumetric merging tends to have numerous tiny triangles. Current graphics processor units (GPUs) present opportunities for real-time rendering of captured data directly. The image from one of the cameras is used to texture each triangle of the model. Figure 9 shows several views of generating novel views using this technique.

We can synthesize more natural view-dependent virtual views by taking advantage of the fact that many cameras have captured the same scene from many angles, and some cameras might be close to the desired virtual view. Instead of assigning one texture per triangle surface element, we can assign multiple textures that come from each of the original camera inputs, normalized and aligned to the surface element. At synthesis time, the system calculates weights for each texture proportionately to the inner product of the new viewing angle and the original camera angle, and it merges the textures with those weights.[21] Figure 7a shows an example input view, and 7b the result. Combined with a better reconstructed 3D model, the synthesized images have much higher quality than those generated using the CSMs and textures. Most of today's 4D digitization and visualization programs, including both ETH and MPI systems, use global models of the scene and render them in similar ways.

Representing dynamic events

Since a dynamic event is a series of temporal snapshots, we can use the correlation between frames for more accurate reconstruction of 4D models. Fitting parametric models might be helpful in certain situations. For instance, we developed a markerless technique to model and track the human body's articulated motion.[20] Another interesting and rarely studied aspect of handling dynamic scenes is temporal interpolation for synthetically resampling the time axis of an event more finely to create a virtual slow motion. Vedula et al.[22] used 3D scene flow and correlation model of points between successive frames to interpolate the event in the time axis.

Manipulating a digitized event

Can we edit and modify the digitized event similar to events created artificially? The manipulation could include adding synthetic objects into it, removing real objects from it, changing the appearance of objects, or transferring the motion of one person to another. If the model's basic representation is compatible with standard graphics models, we can manipulate it using standard tools. Objects created in a computer graphics environment are usually arranged in hierarchies to facilitate efficient handling. The models created from images, however, do not automatically come with the corresponding hierarchical structure. This makes the manipulation of a virtualized event more difficult.

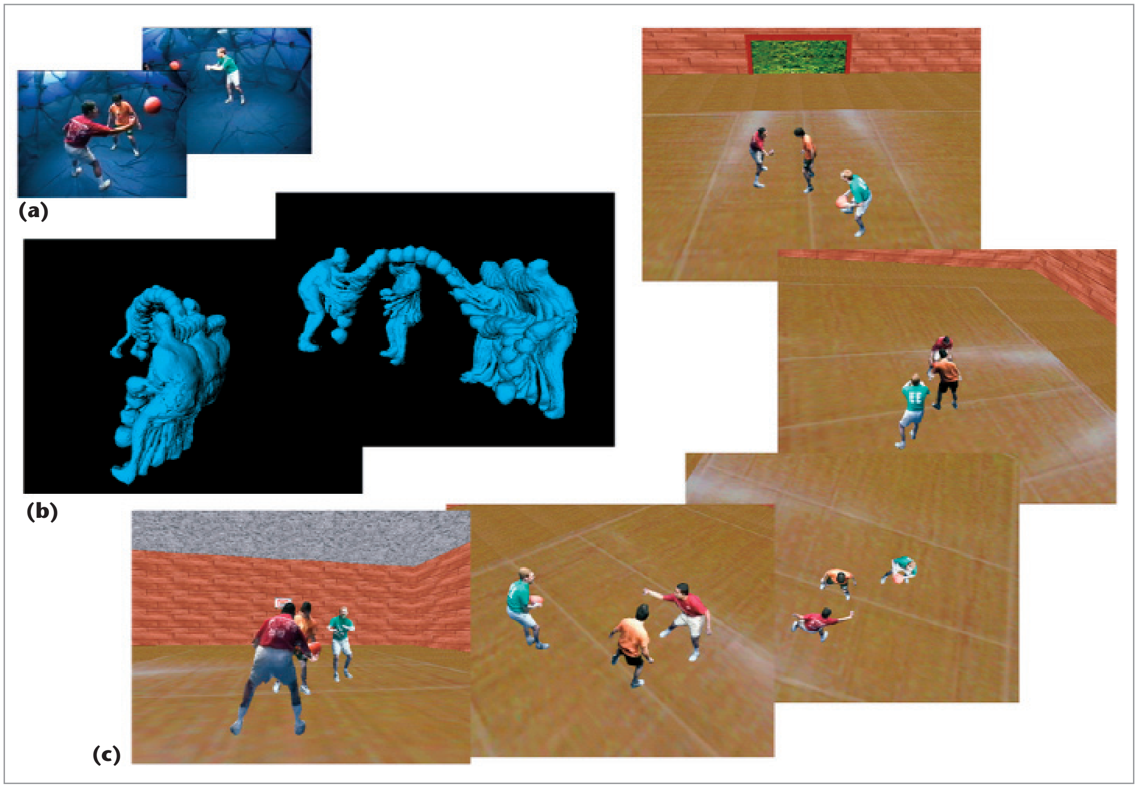
Inserting objects

We can introduce new objects into a virtualized scene and make them interact with real objects. The new objects must be transformed appropriately. Figure 8 shows an early example[2] (circa 1995) of such insertion and interaction in real time by using the real-time stereo machine. In this example, note that the lamp's shadow is cast on the real person consistently with its position and surface shape.

More recent work on appearance editing or relighting involves more accurate modeling and recovering, though not in real time, of material properties and lighting conditions.

Combining multiple events

Once digitized, we can combine multiple independently virtualized events inside the computer into a single interesting event. Figure 9 shows an example.[23] In the one-person event (see Figure 9a), the person throws a ball, and in the two-person event, one person also throws a ball after bouncing it for a while. The two events were combined into a three-person event (see



9 Compositing two virtualized events. (a) One frame each of a one-person event and a two-person event. (b) Time-elapsed version of the model after compositing the two events into a single three-person event. (c) Different views from the dynamic event in an artificial setting.

Figure 9b) in which a ball is thrown from the person in the two-person event to the person in the one-person event. In doing so, the two events were aligned appropriately; the one-person event is time reversed so that he receives the ball instead of throwing it, and the ball trajectory is adjusted accordingly. Figure 9c presents different views in an artificial setting.

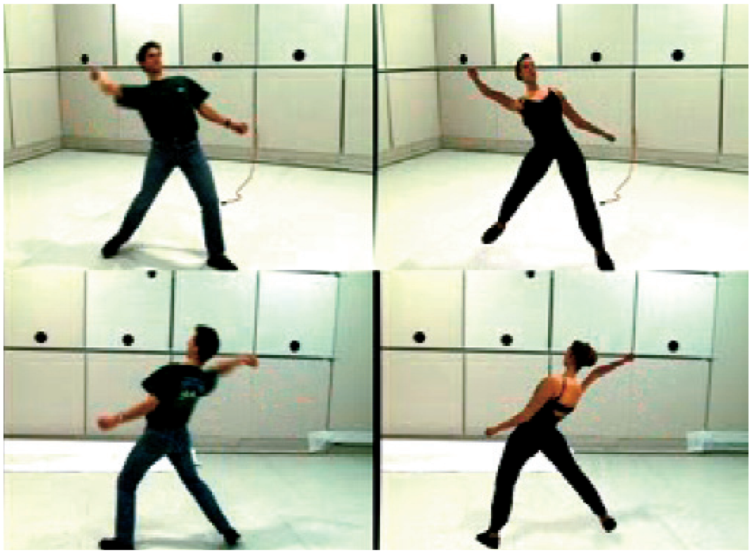
The MSR project also demonstrated compositing different virtualized scenes. They demonstrated taking one ballet and placing it into another by aligning them. The MPI effort is primarily aimed at capturing models of human actors that can subsequently be placed in other virtual environments. Recently, the ETH system developed a tool that can combine multiple events using a video hypercube.[24]

Transferring motion between people

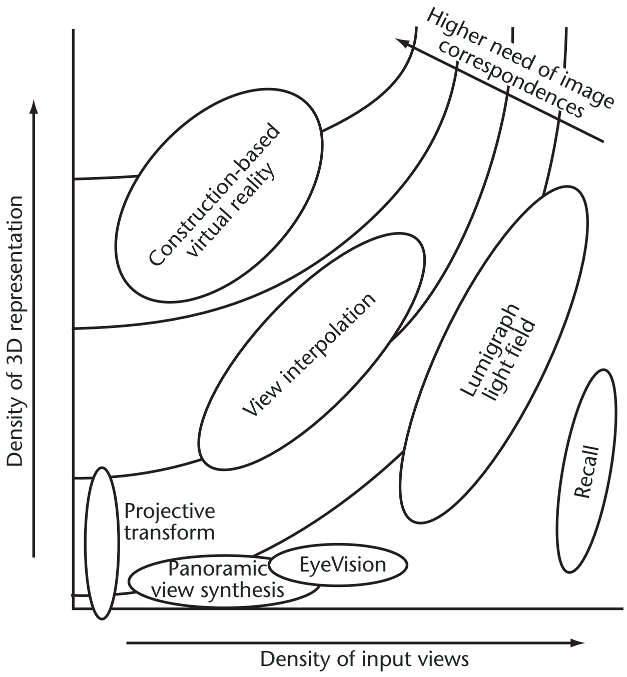
Many systems can capture human motion and drive an avatar or control a humanoid to replicate the motion.[25] These motion-transfer applications use a marker-based motion-capture system; highly reflective markers are attached on limbs, and multiple high-speed cameras track their 3D locations with the help of infrared lighting. We accomplished the same task completely without using markers.[20] In Figure 10, the system first builds the body description (limb lengths, size, appearance of body parts, and articulation kinematics) of the male motion originator from his eight camera views. Then the system analyzes the video of his throwing motion, and without markers it develops his motion description by tracking his motion and matching it with his kinematic model over the video sequence. For motion transfer, the originator's motion drives the articulated body description of the motion recipient (female), which the system has built in the same manner.

"3D-ality" and view density

The techniques for scene digitization and new view synthesis range from those that simply rearrange input images to those that compute the scene's accurate 3D



10 An example of markerless motion transfer. The throw motion of the person (real) on the left transfers to the person (synthetic) on the right.



11 Different digitizing techniques placed in a spectrum of varying density of input views (horizontal axis) and the 3D-ality of the representation (vertical axis). Lightfield or plenoptic function techniques use a large number of views with high density, and employ very little explicit 3D representations of the scene. The Virtualized Reality system, the MSR system, and the ETH system use fewer views, but compute dense and explicit 3D models. View morphing or view interpolation techniques use a relatively small number of input views, and sparse correspondences among them as the 3D representation. Contours running from upper-right to lower-left represent the equi-need of image correspondences—the source of recovering the scene's 3D-ality.

structure. We should understand and compare their strengths and weaknesses from various aspects.

First, different levels of complexity are inherent in the problems that the algorithms deal with. The problem complexity is a function of the "3D-ality" of the world, the input-view density, and the task to be performed. The 3D-ality is the degree to which the world is truly three-dimensional; a planar surface is low in 3D-ality, whereas a porcupine is high. The input-view density is how many and how closely sampled images are given as input. For low 3D-ality problems, simple interpolation-based methods work well, while high 3D-ality problems require either good 3D recovery or high input-view density. Naturally, the 3D-ality and the input-view density offset each other in keeping the complexity in balance. If the scene is not highly 3D, only a small number of input images are required for scene digitization and scene synthesis (except view dependency); a flat planar scene, in fact, can be described using a single image. On the other hand, if the input view density is high enough to effectively cover all possible viewing directions, recalling one of them will do the synthesis job. Most interesting problems are in-between; the number of images is limited, and the scene is fairly 3D. Some tasks are inherently low complexity, even when the world is highly 3D. For example, panoramic view synthesis of images taken from a single view point (or small lateral motion) is a 2D problem. The 3D-ality of the world does not come into play. The EyeVision replay system, though it is an outside-in setup involving a large disparity, is also low in 3D complexity, since the new video does not depart from input views and it relies on human perceptual capability for spatial interpolation.

The second aspect is how a system uses dense or explicit 3D representations internally in various algorithms. Figure 11 shows a mapping of various algorithms, where the horizontal axis is density of input views and the vertical is density of internal 3D representation. Image-based rendering techniques using plenoptic functions, such as a Lightfield[8] and a Lumigraph,[7] used thousands of views with high density, and employ few explicit 3D representations of the scene. Instead, the approximated plenoptic functions represent relationships between 3D-ality and views in a compiled manner. On the other hand, the Virtualized Reality system, the MSR system, and the ETH system use fewer views, but compute dense and explicit 3D models using multiview computer vision techniques. Other image-based rendering techniques, such as view morphing or view interpolation, use a relatively small number of input views, and the scene's 3D-ality is represented by sparse correspondences among them. This map reveals a common misconception that image-based rendering methods do not use 3D models; they represent them implicitly. The set of image correspondences among images *is* the model. The sparseness is relying on the assumption that the world is piecewise planar. In the case of Lumigraph, the placement of the *u*-*v* plane (typically at the average depth) represents the approximation that depth variation within the object is not significant. As the 3D-ality of the world increases, either their representation density must increase (for example, the density of correspondences increases or the *u*-*v* plane is replaced by a piece-wise planar surface to better approximate the shape) or the input density must increase.

The third and least explored aspect is extrapolating the virtualized world beyond the envelope of the given images. Extrapolation (as well as manipulation) of the virtualized world often requires explicit modeling of the scene's geometric and photometric content, including objects (preferably separated), motion, material, and lighting, so that we can manipulate them individually. Increasing the view density alone does not solve the extrapolation problem automatically, and most of the algorithms coming from computer graphics community require off-line processing. Recent work on computational cameras that recover object geometry and material property simultaneously by separating effects of camera's own active lighting such as flash and those of ambient lighting is promising.[26] Ultimate Virtualized Reality must have the capability for such a high-degree of abstraction and extrapolation.

Conclusions

Capturing a static or dynamic scene from the real world into the computer for synthesizing its new images or video has been an active research area in the past decade. However, Virtualized Reality's original goal is still a long way from being realized. Multicamera modeling and digitization of a large dynamic scene continue to be important and challenging problems as their application scenarios expand.

Choice of representations for dynamic scenes remains a critical issue. Representations based on depth images are natural, easy to render with good quality due to the locality properties, and amenable to good compression. Global models, while convenient, are difficult to compute robustly. Point-based representations, including ordered-point representation, might become important for representing digitized dynamic events. Hybrid representations that use approximate or proxy geometry and good texture are also promising; dynamic light fields and depth-image-based rendering extend IBR to dynamic scenes.

Camera resolution and quality also continue to improve. However, stereo and image-correspondence algorithms must keep pace with the sensor improvement. Algorithms that combine motion, silhouettes, shading, and other information to recover the global structure will produce the better structure and appearance model of the scene.

Finally, recovery of the input scene's lighting parameters and each surface's accurate material properties must improve considerably if the digitization of dynamic events moves into the realm of true abstraction and extrapolation. ■

References

1. T. Kanade et al., "Development of a Video-Rate Stereo Machine," *Proc. Int'l Robotics and Systems Conf.* (IROS), IEEE CS Press, vol. 3, 1995, pp. 95-100.
2. T. Kanade et al., *Video-Rate Z Keying: A New Method for Merging Images*, tech. report CMU-RI-TR-95-38, Robotics Institute, Carnegie Mellon University, 1995.
3. T. Kanade, P. Rander, and P.J. Narayanan, "Virtualized Reality: Constructing Virtual Worlds from Real Scenes," *IEEE Multimedia*, vol. 4, no. 1, 1997, pp. 34-47.
4. P.J. Narayanan, P.W. Rander, and T. Kanade, "Constructing Virtual Worlds Using Dense Stereo," *Proc. IEEE Int'l Conf. Computer Vision* (ICCV), IEEE CS Press, 1998, pp. 3-10.
5. S. Moezzi, L.-C. Tai, and P. Gerard, "Virtual View Generation for 3D Digital Video," *IEEE Multimedia*, vol. 4, no. 1, 1997, pp. 18-26.
6. P.E. Debevec, C.J. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry-and Image-Based Approach," *Proc. 23rd Ann. Conf. Computer Graphics and Interactive Techniques*, ACM Press, 1996, pp. 11-20.
7. S.J. Gortler et al., "The Lumigraph," *Proc. 23rd Ann. Conf. Computer Graphics and Interactive Techniques*, ACM Press, 1996, pp. 43-54.
8. M. Levoy and P. Hanrahan, "Lightfield rendering," *Proc. 23rd Ann. Conf. Computer Graphics and Interactive Techniques*, 1996, pp. 31-42.
9. G. Godin et al., "Active Optical 3D Imaging for Heritage Applications," *IEEE Computer Graphics and Applications*, vol. 22, no. 5, 2002, pp. 24-36.
10. M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. 27th Ann. Conf. Computer Graphics and Interactive Techniques*, ACM Press, 2000, pp. 131-144.
11. T. Oishi et al., "Digital Restoration of the Original Great Buddha and Main Hall of Todaiji Temple," *Trans. Virtual Reality Soc. Japan*, vol. 10, no. 3, 2005, pp. 429-436.
12. M. Gross, et al., "Blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence," ACM Trans. Graphics, vol. 22, no. 3, 2003, pp. 819-827.
13. S. Wurmlin et al., "3D Video Recorder," *Pacific Conf. Computer Graphics and Applications*, IEEE CS Press, 2002, pp. 325-334.
14. J. Carranza et al., "Free-Viewpoint Video of Human Actors," *ACM Trans. Graphics*, vol. 22, no. 3, 2003, pp. 569-577.
15. C.L. Zitnick, "High-Quality Video View Interpolation Using a Layered Representation," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 600-608.
16. M. Tanimoto, "Free Viewpoint Television for 3D Scene Reproduction and Creation," *Proc. Conf. Computer Vision and Pattern Recognition Workshop* (CVPRW), IEEE CS Press, 2006, p. 172; http://www.tanimoto.nuee.nagoya-u.ac.jp.
17. P.J. Narayanan, P.W. Rander, and T. Kanade, *Synchronizing and Capturing Every Frame from Multiple Cameras*, tech. report CMU-RI-TR-95-25, Robotics Institute, Carnegie Mellon University, 1995.
18. 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. 3, no. 4, 1987, pp. 323-344.
19. Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, 2000, pp. 1330-1334.
20. G.K.M. Cheung, S. Baker, and T. Kanade, "Shape-from-Silhouette Across Time Part II: Applications to Human Modeling and Markerless Motion Tracking" *Int'l J. Computer Vision*, vol. 63, no. 3, 2005, pp. 225-245.
21. H. Saito, S. Baba, and T. Kanade, "Appearance-Based Virtual View Generation from Multicamera Videos Captured in the 3D Room, *IEEE Trans. Multimedia*, vol. 5, no. 3, 2003, pp. 303-316.
22. S. Vedula, S. Baker, and T. Kanade, "Image-Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events," *ACM Trans. Graphics*, vol. 24, no. 2, 2005, pp. 240-261.
23. T. Kanade et al., "Virtualized Reality: Digitizing a 3D Time-Varying Event as is and in Real Time," *Mixed Reality: Merging Real and Virtual Worlds*, Y. Ohta and H. Tamura, eds., Ohmsha, Ltd. and Springer, 1999, pp. 41-57.
24. M. Waschbusch, S. Wurmlin, and M. Gross, "Interactive 3D Video Editing," *The Visual Computer*, vol. 22, no. 9, 2006, pp. 631-641.
25. M. Oshita, "Motion-Capture-Based Avatar Control Framework in Third-Person View Virtual Environments," *Proc. ACM SIG Computer-Human Interaction Int'l Conf. Advances in Computer Entertainment Technology*, ACM Press, 2006, art. no. 2.
26. S. K. Nayar, "Computational Cameras: Redefining the Image," *Computer*, vol. 39, no. 8, 2006, pp. 30-38.

Takeo Kanade is the U.A. and Helen Whitaker University Professor of Computer Science and Robotics at Carnegie Mellon University and the director of Quality of Life Technology Engineering Research Center. He is also director of Digital Human Research Center in Tokyo. He has a PhD in electrical engineering from Kyoto University. His research interests include multiple areas of robotics: computer vision, multimedia, manipulators, autonomous mobile robots, medical robotics and sensors. He is an IEEE Fellow, an ACM Fellow, and a Founding Fellow of American Association of Artificial Intelligence (AAAI). Contact him at tk+@cs.cmu.edu.

P.J. Narayanan is a Professor and Dean of Research at the International Institute of Information Technology, Hyderabad. His research interests include computer vision, computer graphics, and virtual reality. He has a PhD in computer science from the University of Maryland. He was the general chair of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2000) and program chair of the Asian Conference on Computer Vision (ACCV 2006). Contact him at pjn@iiit.ac.in.