Presented at IEEE Workshop on the Representation of Visual Scenes, Boston, June 24, 1995.

Virtual*ized* Reality: Concepts and Early Results

Takeo Kanade, P. J. Narayanan, Peter W. Rander

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The visual medium evolved from early paintings to the realistic paintings of the classical era to photographs. The medium of moving imagery started with motion pictures. Television and video recording advanced it to show action "live" or capture and playback later. In all of the above media, the view of the scene is determined at the transcription time, independent of the viewer.

We have been developing a new visual medium called virtualized reality. It delays the selection of the viewing angle till view time, using techniques from computer vision and computer graphics. The visual event is captured using many cameras that cover the action from all sides. The 3D structure of the event, aligned with the pixels of the image, is computed for a few selected directions using a stereo technique. Triangulation and texture mapping enable the placement of a "soft-camera" to reconstruct the event from any new viewpoint. With a stereo-viewing system, virtualized reality allows a viewer to move freely in the scene, independent of the transcription angles used to record the scene.

Virtualized reality has significant advantages over virtual reality. The virtual reality world is typically constructed using simplistic, artificially-created CAD models. Virtualized reality starts with the real world scene and virtualizes it. It is a fully 3D medium as it knows the 3D structure of every point in the image.

The applications of virtualized reality are many. Training can become safer and more effective by enabling the trainee to move about freely in a virtualized environment. A whole new entertainment programming can open by allowing the viewer to watch a basketball game while standing on the court or while running with a particular player. In this paper, we describe the hardware and software setup in our "studio" to make virtualized reality movies. Examples are provided to demonstrate the effectiveness of the system.

1 Introduction

We have a few visual media available today: paintings, photographs, moving pictures, television and video recordings. They share one aspect: the view of the scene is decided by a "director" while recording or transcribing the event, independent of the viewer.

We describe a new visual medium called *virtualized reality*. It delays the selection of the viewing angle till *view time*. To generate data for such a medium, we record the events using many cameras, positioned so as to cover the event from all sides. The time-varying 3D structure of the event, described in terms of the depth of each point and aligned with the pixels of the image, is computed for a few of the camera angles — called the *transcription angles* — using a stereo method. We call this combination of depth and aligned intensity images the *scene description*. The collection of a number of scene descriptions, each from a different transcription angle is called the *virtualized* world. Once the real world has been virtualized, graphics techniques can render the event from any viewpoint. The scene description from the transcription angle closest to the viewer's position can be chosen dynamically for rendering by tracking the position and orientation of the viewer. The viewer, wearing a stereo-viewing system, can freely move about in the world and observe it from a viewpoint chosen dynamically at *view time*.

Virtualized reality improves traditional virtual reality. Virtual reality allows viewers to move in a virtual world but lacks fine detail as their worlds are usually artificially created using simplistic CAD models. Virtualized reality, in contrast, starts with a real world and virtualizes it.

There are many applications of virtualized reality. Training can become safer and more effective by enabling the trainee to move about freely in a virtualized environment. A surgery, recorded in a virtualized reality studio, could be revisited by medical students repeatedly, viewing it from positions of their choice. Telerobotics maneuvers can be rehearsed in a virtualized environment that feels every bit as real as the real world. True telepresence could be achieved by performing transcription and view generation in real time. And an entirely new generation of entertainment media can be developed: basketball enthusiasts and broadway aficionados could be given the feeling of watching the event from their preferred seat, or from a seat that changes with the action.

Stereo or image-matching methods, which are the key components in virtualized reality, are well-studied. Precise reconstruction of the whole scene using a large number of cameras is, however, relatively new. Kanade[6] proposed the use of multi-camera stereo using supercomputers for creating 3D models to enrich the virtual world. Rioux, Godin and Blais[16] outlined a procedure to communicate complete 3D information about an object using depth and reflectance. Fuchs and Neuman[3] presented a proposal to achieve telepresence for medical applications. Some initial experiments were conducted at CMU using the video-rate stereo machine[7][9], by the team of UNC, UPenn and CMU[2], and at Tsukuba by Ohta and Satoh[10]. Laveau and Faugeras[8] attempt “view transfer” with uncalibrated cameras using epipolar constraints alone.

This paper introduces the concept of virtualized reality. We present the three stages of creating a virtualized real scene — scene transcription, structure extraction and view generation — in the next three sections. Early examples from the virtualized reality studio are interspersed with the discussion to elucidate the concepts. Our experimental setup at present consists of a 5m dome, 10 cameras and VCRs, a digitizing setup, several workstations and a Silicon Graphics Onyx RE2 graphics workstation.

2 Scene Transcription

The central idea of this research is that we can virtualize real-world scenes by capturing *scene descriptions* — the 3D structure of the scene aligned with its image — from a number of *transcription angles*. The scene can be synthesized from any viewpoint using one or more scene descriptions. The facility to acquire the scene descriptions is called the *virtualized reality studio.* Any such studio should cover the action from all angles. Stereo techniques used to extract the scene structure require images corresponding to precisely the same time instant from every camera to be fed to them in order to accurately recover 3D scene structure. We potentially need to virtualize every frame in video streams containing fast moving events to satisfactorily reproduce the motion. Therefore, the studio should have the capability to record and digitize every frame of each video stream synchronously. We elaborate on the physical studio, the recording setup and the digitizing setup in this section.

2.1 The Studio Setup

Figure 1(a) shows the studio we have in mind. Cameras are placed all around the dome, providing views from angles surrounding the scene. Figure 1(b) show the studio we have built using a hemispherical dome, 5 meters in diameter, constructed from nodes of two types and rods of two lengths. We currently have 10 cameras — 2 color cameras and 8 monochrome ones — to transcribe the scene. We typically arrange them in two clusters, each providing a scene description, with the transcription angles given by the color cameras. The cameras are mounted on special L-shaped aluminum brackets that can be clamped on anywhere on the rods.

2.2 Synchronous Multi-camera Recording

To synchronously record a set of cameras, a single control signal could be supplied to the cameras to simultaneously acquire images and to the video recording equipment to simultaneously store the images. In order to implement this approach directly in digital recording hardware, the system would need to handle the real-time video streams from each camera. For a single color camera providing 30 images per second, 512x512 pixels per image, 3 color bands per pixel, and 8 bits per color band, the system would need to handle 22.5 MBytes of image data per second. Even if loss-less compression could reduce this bandwidth by a factor of 3, a small, a 10-camera system would need a sustained 75 MBytes per second of bandwidth in addition to real-time image compression. Typical image capture and digital storage systems, however, fall far short of providing this capacity. For example, our current system — a Sun Sparc 20 workstation with a K^2T V300 digitizer — can capture and store only about 750 KBytes per second. Specialized hardware could improve the throughput but at a substantially higher cost.

We developed an off-line system to synchronously record frames from multiple cameras. The cameras are first synchronized to a common sync signal. The output of each

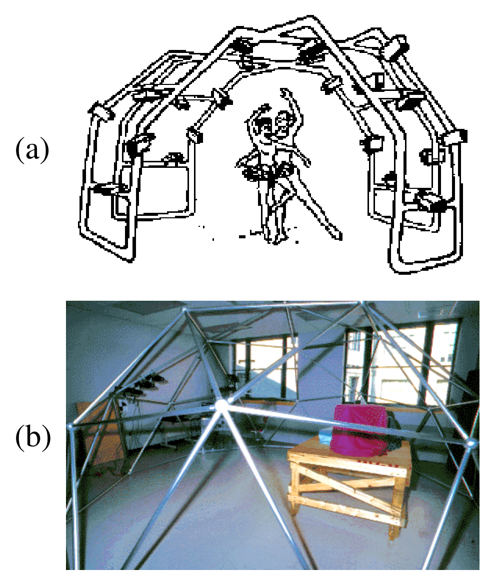


Figure 1: The virtualized reality studio.
(a) Conceptual. (b) The dome.

camera is time stamped with a common Vertical Interval Time Code (VITC) and recorded on tape using a separate VCR. The tapes are digitized individually off-line using a frame grabber and software that interprets the VITC time code embedded in each field. We can capture all frames of a tape by playing the tape as many times as the speed of the digitizing hardware necessitates. The time code also allows us to correlate the frames across cameras, which is crucial when transcribing moving events. Interested readers can refer to a separate report[17] for more details on the synchronous multi-camera recording and digitizing setup. Figure 2 shows a still frame as seen by six cameras of the virtualizing studio digitized using the above setup.

3 Structure Extraction

We use the multi-baseline stereo (MBS) technique [9] to extract the 3D structure from the multi-camera images collected in our virtualized reality studio. Stereo algorithms compute estimates of scene depth from correspondences among images of the scene. The choice of the MBS algorithm was motivated primarily by two factors. First, MBS recovers dense depth maps — that is, a depth estimate corresponding to every pixel in the intensity images — which is needed for image reconstruction. Second, MBS takes advantage of the large number of cameras that we are using for scene transcription to increase precision and reduce errors in depth estimation.

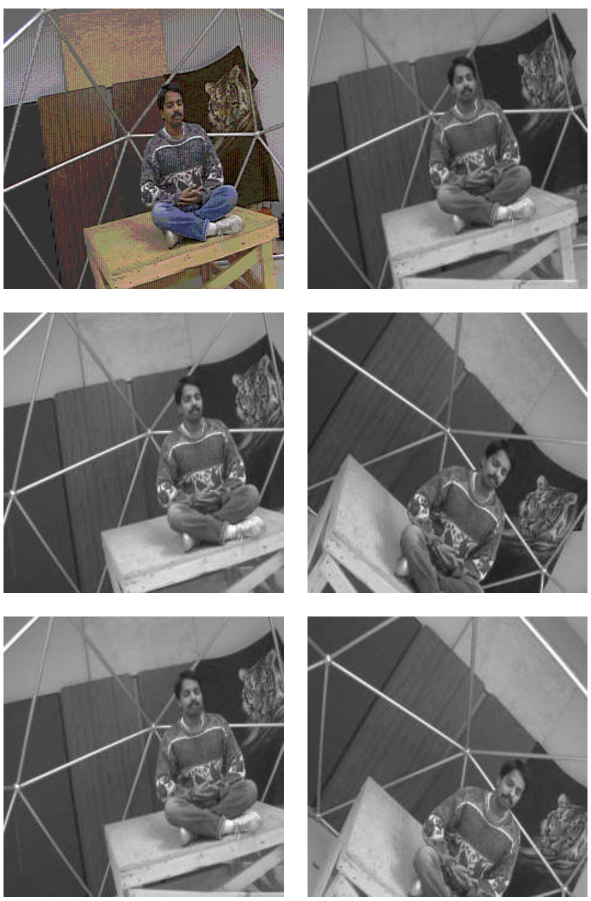


Figure 2: A scene as seen by six cameras.

3.1 Fundamentals of Multi-Baseline Stereo

To understand the MBS algorithm, consider a multi-camera imaging system in which the imaging planes of the cameras all lie in the same physical plane and in which the cameras have the same focal length F. For any two of the cameras, the disparity d (the difference in the positions of corresponding points in the two images) and the distance z to the scene point are related by

$$d = BF\frac{1}{z}$$

where B is the baseline, or distance between the two camera centers. The simplicity of this relation makes clear one very important fact: the precision of the estimated distance increases as the baseline between the cameras increases. In theory, the cameras can be placed as far apart as possible. Practical experience using stereo systems reveals, however, that increasing the baseline also increases the likelihood of mismatching points among the images. There is a trade-off between the desires for correct correspondence among images (using narrow baselines) and for precise estimates of scene depth (using wide baselines).

The multi-baseline stereo technique attempts to eliminate this trade-off by simultaneously computing correspondences among pairs of images from multiple cameras with multiple baselines. In order to relate correspondences from multiple image pairs, we rewrite the previous equation as

$$\frac{d}{BF} = \frac{1}{z} = \varsigma$$

which indicates that for any point in the image, the inverse depth (ς) is constant since there is only one depth z for that point. If the search for correspondences is computed with respect to ς, it should consistently yield a good match at the correct value of ς independently of the baseline B. With multiple (more than 2) cameras, correspondences can now be related across camera pairs, since the searching index ς is independent of the baselines. The resulting correspondence search combines the correct correspondence of narrower baselines with the higher precision of wider baselines, and it has been proven that it yields a unique match of high precision [9].

One way to find 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 correct position of the window in the second image is constrained by the camera geometry to lie along the epipolar line of the position in the first image. The matching process involves shifting the window along this line as a function of ς, computing the match error — using normalized correlation or sum of squared differences (SSD) — over the window at each position, and finding the minimum error. The estimate of inverse depth, $\hat{\varsigma}$, is the ς at this minimum.

To demonstrate the advantages of multi-baseline stereo, consider the data presented in Figure 3. Part (a) shows match error as a function of ς for 3 camera pairs. In this set of cameras, we see both of the problems previously discussed: poor localization (in the top curve) for a shorter baseline and false minima (in the bottom curve) for a longer baseline. Applying the multi-baseline stereo algorithm to this data yields the error curve in Figure 3(b). This curve has only the single minimum at the correct location with a sharp profile.

3.2 Depth Map Editing

Window-based correspondence searches suffer from a well-known problem: inaccurate depth recovery along depth discontinuities and in regions of low image texture. The recovered depth maps tend to "fatten" or "shrink" objects along depth discontinuities. This phenomena occurs because windows centered near the images of these discontinuities will contain portions of objects at two different depths. When one of these windows is matched to different images, one of two situations will occur. Either the foreground object will occlude the background object so that depth estimates for the background points will incorrectly match to the portion of the foreground in the window, or both the foreground and background regions will remain visible, leading to two likely candidate correspondences. In regions with little texture — that is, of fairly constant intensity — window-based correspondence searches yield highly uncertain estimates of depth. Consider, for example, a stereo image pair with constant intensity in each image. With no intensity variation, any window matches all points equally well, making any depth estimates meaningless.

To address this inaccuracy in depth recovery, we could reduce the window size used during matching, potentially matching individual pixels. This approach reduces the number of pixels effected by depth discontinuities. By doing so, however, we also reduce the amount of image texture contained within the window, increasing the uncertainty of the recovered depth estimate. Conversely, we could increase the size of the window to give more image texture for matching. This action increases the image texture contained in the window, but also increases the area effected by the discontinuities. Optimizing the window size requires trading off the effects of the depth discontinuities with those of the low-texture regions.

In order to work around this trade-off, we have incorporated an interactive depth map editor into our process of structure extraction. Rather than send the MBS-computed depth maps directly on to the next processing stage, we instead manually edit the depth map to correct the errors that occur during automatic processing. While a good window size still helps by reducing the number of errors to be corrected, it is less important in this approach because the user can correct the problems in the depth maps. We are currently exploring modifications to the stereo algorithm in an effort to reduce or eliminate this need for human intervention.

3.3 General Camera Configurations

For general camera positions, we perform both intrinsic and extrinsic camera calibration to obtain epipolar line constraints, using either an approach from Szeliski and Kang [13] or one from Tsai [14]. Using the recovered calibration, any point in the 3D coordinate system of the reference camera can be mapped to a point in the 3D coordinate system of any of the other cameras. To find correspondences, we again match a reference region to another image as a function of inverse depth ς. To find the position in the second image corresponding to this inverse depth, we convert the reference point and inverse depth into a 3D coordinate, apply the camera-to-camera mapping, and project the converted 3D point into the other image. As with the parallel-camera configuration, the full search is conducted by matching each reference image point to the other images for each possible ς. We then add the match error curves from a set of image pairs and search for the minimum of the combined error function. Figure 3 (c) shows the depth map recovered by applying this approach to the input images shown in Figure 2. The depth map has 74 levels for a depth range of 2 meters to 5 meters.

4 View Generation

We described how to "virtualize" an event in terms of a number of scene descriptions in the previous sections. The medium of virtualized reality needs to synthesize the scene from arbitrary viewpoints using these scene descriptions. To render the scene from other viewpoints using graphics workstations, we translate the scene description into an object type, such as a polygonal mesh. We texture map an intensity image onto the rendered polygons, generating visually realistic images of the scene. Graphics workstations have specialized hardware to render them quickly. A Silicon Graphics Onyx/RE2 can render close to 1 million texture mapped triangles per second.

We describe how new views are generated from a single scene description first. The generated view will be lower in quality as the viewpoint gets far from the transcription angle. We discuss how we can use multiple scene descriptions to get realistic rendering from all angles.

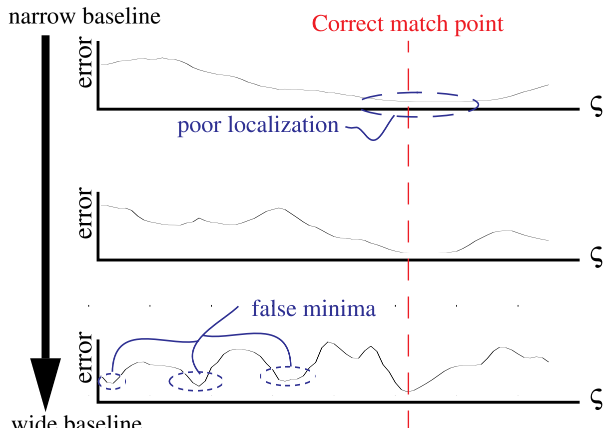
4.1 Using a Single Scene Description

A scene description consists of a depth map providing a dense three dimensional structure of the scene aligned with the intensity map of the scene. The point (*i*, *j*) in the depth map gives the distance of the intensity image pixel (*i*, *j*) from the camera. We convert the depth map into a triangle mesh and the intensity map to texture to render new views on a graphics workstation. There are two aspects of performing this translation realistically: object definition and occlusion handling.

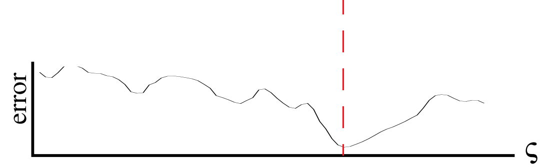
4.1.1 Object Definition. Graphics rendering machines synthesize images of a scene from an arbitrary point of view given a polygonal representation of the scene. Texture mapping pastes an intensity image onto these rendered polygons, generating visually realistic images of the scene from arbitrary view points. We currently generate a triangle mesh from the depth map by converting every 2×2 section of the depth map into two triangles. Figure 4 illustrates how the mesh is defined. The *(x, y, z)* coordinates of each point in the image are computed from the image coordinates and the depth, using the intrinsic parameters of the imaging system. Each vertex of the triangle also has a texture coordinate from the corresponding intensity image. This simple method results in $2 \times (m-1) \times (n-1)$ triangles for a depth map of size $m \times n$. The number of triangles for the depth map shown in Figure 3 is approximately 200,000. Though this is a large number of triangles, the regularity makes it possible to render them efficiently on graphics workstations.

We reduce the number of triangles in our scene definition by adapting an algorithm developed by Garland and Heckbert that simplifies a general dense elevation/depth map into planar patches[4]. The algorithm computes a triangulation using the smallest number of vertices given a measure for the maximum deviation from the original depth map. The procedure starts with two triangles defined by the outer four vertices. It repeatedly grows the triangle mesh by adding the vertex of maximum deviation and the corresponding triangle edges till the maximum deviation condition is reached. Using this technique, we have reduced mesh size by factors of 20 to 25 on typical scenes without affecting the visual quality of the output.

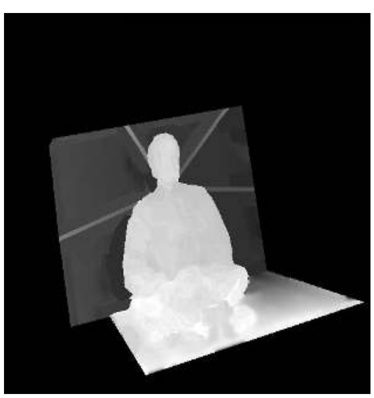
4.1.2 Occlusion Handling. The simple rendering technique described above treats the entire depth map as one



(a) Match error for camera pairs.

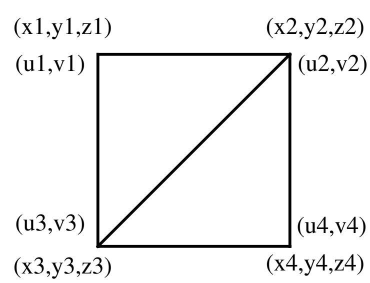


(b) Match error for multi-baseline technique.



(c) Depth map for images in Figure 2.

Figure 3: Results of multi-baseline stereo algorithm.



Triangle 1:

Vertex 1: (x1,y1,z1) texture coord: (u_1/m , v_1/n)
Vertex 2: (x2,y2,z2) texture coord: (u_2/m , v_2/n)
Vertex 3: (x3,y3,z3) texture coord: (u_3/m , v_3/n)

Triangle 2:

Vertex 1: (x2,y2,z2) texture coord: (u_2/m , v_2/n)
Vertex 2: (x3,y3,z3) texture coord: (u_3/m , v_3/n)
Vertex 3: (x4,y4,z4) texture coord: (u_4/m , v_4/n)

Figure 4: Triangle mesh and texture coordinate definition.

large surface, connecting pixels across depth discontinuities at object boundaries. This introduces an artificial surface bridging the discontinuity, with the few pixels of texture stretched over the surface. When generating views for angles far from the transcription angle, these surfaces become large and visually unrealistic; in Figure 5(a), for instance, the person and the wall appear to be connected. We therefore delete these artificial surfaces by not rendering the triangles that overlap discontinuities, resulting in "holes" as seen in Figure 5(b). We fill these holes using other scene descriptions as explained in Section 4.2.

4.1.3 Experimental Results. Figure 6 shows how the example scene appears from a few different viewpoints when rendered with this technique. While still not completely realistic — we expect to see something behind the man — the visual realism of this method is far superior to the initial approach.

The discussion to this point has focussed on virtualizing a single, static scene. It is also possible to virtualize moving scenes by virtualizing each frame separately. The resulting virtualized reality movie can be played with the viewer standing still anywhere in the world by rendering each frame from the viewer's position. The scene can also be observed by a viewer whose movement through the world is independent of the motion in the scene. Figure 7 shows seven frames of a basketball sequence from the reference transcription point and from a synthetically-created moving viewpoint.

4.2 Merging Multiple Scene Descriptions

There are two reasons for combining the scene descriptions from multiple transcription angles while generating new views. First, as discussed in Section 4.1, depth discontinuities appear as holes in views far from the transcription angle when using a single scene description. We should "fill" these holes using a scene description from another transcription angle for which the portion of the scene is not occluded. Second, the intensity image used for texturing gets compressed or stretched when the viewing angle is far from the transcription angle, resulting in poor quality of the synthesized image. If the viewer strays far from the starting position, we should choose the most direct transcription angle for each viewing angle to minimize this degradation.

Filling the gaps requires some thought. In a well calibrated ideal world, the scene descriptions computed from different transcription angles will match one another exactly. In practical systems, however, they will be misaligned somewhat due to the errors in calibration and matching. The triangle vertices that fall on the gap contours will not exactly match in two transcription angles. These factors complicate view merging.

(a)

(b)

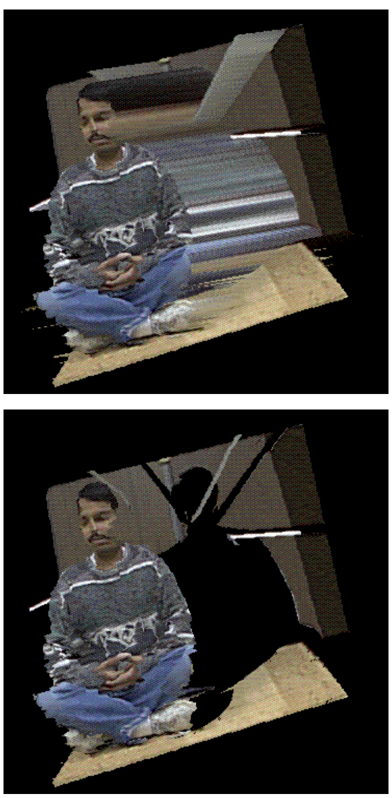


Figure 5: (a) View without discontinuity compensation. (b) With compensation.

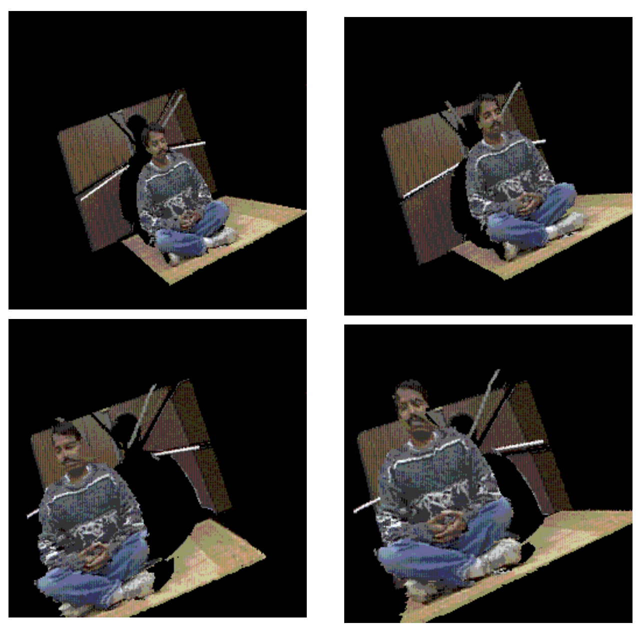


Figure 6: The scene from four alternate viewpoints.

(a) Original reference images.

(b) Synthesized from a moving viewpoint. The starts left and above the original transcription view and moves to the right.

Figure 7: Seven frames of a basketball sequence.

One strategy is to combine the scene descriptions from all transcription angles ahead of time to generate a model of the scene that contains all the necessary detail. Several methods are available to register and model objects from multiple range images[5][11][12][15]. Such a consolidated model attempts to give one grand description of the entire world. We only require the best partial description of the world visible from a particular viewing angle at any time. Such a partial description is likely to be more accurate due to its limited scope; inaccuracies in the recovery of the portion not seen will not affect it. It is likely to be simpler than a consolidated model of the scene, lending easily to real time view generation. The partial description we use consists of a reference scene description from the transcription angle closest to the viewing angle plus one or two supporting ones. The reference description is used for rendering most of the view and the supporting ones are used for filling the gaps.

For combining the views from the reference and the supporting scene descriptions, we currently use the simple strategy of combining at image level. We render the image for the same viewing angle using the reference transcription angle and the supporting ones. The holes in the rendering of the reference description are filled directly with the pixel values from the rendering of another. Figure 8 shows a scene rendered from the same view point using the reference scene description and using a supporting scene description. It also shows the result of combining them at the image level. This method holds promise when the transcription angles are fairly densely distributed.

5 Conclusions

We introduced and elaborated on the concept of virtualized reality in this paper. It combines techniques from computer vision and computer graphics to *virtualize* a real

(a) Reference view.

(b) Supporting view.

(c) Merged view.

Figure 8: Merging images from two scene descriptions.

world event and to let a viewer move about freely in the virtualized world. We also demonstrated the efficacy of virtualized reality using scenes virtualized in our studio to make such movies. It is today possible to virtualize an event such as a surgery and let trainees move about it in a realistic recreation of the surgery in a manner they prefer. We plan to push the training and entertainment applications of virtualized reality in the future.

Acknowledgments: We would like to thank Richard Szeliski and Sing Bing Kang for the use of their calibration program. We also acknowledge the use of Reg Wilson's calibration program.

6 References

[1] M. J. Frankel and J. A. Webb. Design, Implementation and Performance of a Multi-Camera Interactive Video Display System. *Computer Architectures for Machine Perception*. To appear. Como, Italy, 1995.

[2] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S.W. Lee, H. Farid, and T. Kanade. Virtual Space Teleconferencing using a Sea of Cameras, In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, pp.161-167, 1994

[3] H. Fuchs and U. Neuman. A Vision Telepresence for Medical Consultation and other Applications. In *Sixth International Symposium of Robotics Research*, pages 555–571, 1993.

[4] M. Garland and P. Heckbert. Terrain Simplification. *In Preparation.*

[5] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, Piecewise Smooth Surface Reconstruction, *Computer Graphics SIGGRAPH'94*, 295-302, 1994.

[6] T. Kanade. User Viewpoint: Putting the Reality into Virtual Reality. *MasPar News*, 2(2), Nov. 1991.

[7] T. Kanade. Very Fast 3-D Sensing Hardware. In *Sixth International Symposium of Robotics Research*, pages 185–198, 1993.

[8] S. Laveau and O. Faugeras. 3-D Scene Representation as a Collection of Images and Fundamental Matrices, *INRIA Tech Report* 2205, 1994.

[9] M. Okutomi and T. Kanade. A multiple-baseline stereo, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353-363, 1993.

[10]K. Satoh and Y. Ohta. Passive Depth Acquisition for 3D Image Displays. In *IEICE Transactions on Information and Systems*, E77-D(9), Sep. 1994.

[11]H. Shum, An Integral Approach to Free-form Object Modelling, *Carnegie Mellon University Tech Report* CMU-CS-95-135, 1995.

[12]M. Soucy and D. Laurendeau, Multi-Resolution Surface Modelling from Multiple Range Views, *Proceedings of IEEE CVPR'92*, 348-353, 1992.

[13]R. Szeliski and S.B. Kang, Recovering 3D Shape and Motion from Image Streams using Non-Linear Least Squares, *Journal of Visual Communication and Image Representation*, 5(1):10-28, 1994

[14]R. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses, *IEEE Journal of Robotics and Automation*, RA-3(4):323-344, 1987.

[15]G. Turk and M. Levoy, Zippered Polygon Meshes from Range Images, *Computer Graphics SIGGRAPH'94*, 311-318, 1994.

[16]M. Rioux, G. Godin, and F. Blais, Datagraphy: The Final Frontier in Communications, In *International Conference on Three Dimensional Media Technology*, 1992.

[17]P. J. Narayanan, P. Rander, and T. Kanade, Synchronizing and Capturing Every Frame from Multiple Cameras, Robotics Technical Report, CMU-RI-TR-95-25, 1995.