CODING APPROACHES FOR END-TO-END 3D TV SYSTEMS

Anthony Vetro, Wojciech Matusik, Hanspeter Pfister, Jun Xin

Mitsubishi Electric Research Laboratories, Cambridge, MA

ABSTRACT

We present a 3D TV prototype system with real-time acquisition, transmission and auto-stereoscopic display of dynamic scenes. Our system uses a distributed, scalable architecture to manage the high computation and bandwidth demands. It consists of an array of cameras, clusters of network-connected PCs, and a multi-projector 3D display. The 3D display shows high-resolution (1024×768) stereoscopic color images for multiple viewpoints without special glasses. We implemented systems with rear-projection and front-projection lenticular screens. In this paper, we provide an overview of our 3D TV system, including an examination of design choices and tradeoffs. We also discuss potential coding approaches for multiple view video, such as simulcasting, spatio-temporal encoding and sampling-based methods.

1. INTRODUCTION

Three-dimensional TV is expected to be the next revolution in the history of television. Despite some research in the early 1960's, end-to-end 3D TV systems were not technically and commercially viable until recently. Although there are many factors involved, 3D TV requires capturing and displaying multi-view video of real-life dynamic scenes, preferably in real-time. Unfortunately, the high processing and bandwidth requirements of end-to-end 3D TV exceed the capabilities of most systems. To address these issues, we have built a prototype 3D TV system with the following features:

- **End-to-End 3D TV**: Except for broadcasting over a digital channel, our system implements all aspects of 3D TV, including multi-view video acquisition, compression, and 3D display.
- **Distributed Architecture**: We use a distributed clusters of PCs to handle the large processing and bandwidth requirements of multi-view video.
- **Real-Time Performance**: Arbitrary scenes can be acquired and displayed in real-time with only a minimal amount of lag.
- **Scalability**: The system is completely scalable in the number of acquired, transmitted, and displayed views.
- **Projection-based 3D display**: Our 3D display uses an array of 16 projectors to provide a high-resolution display with $16 \times 1024 \times 768$ pixels. A lenticular screen provides auto-stereoscopic images with horizontal parallax and 16 views.
- **Computational alignment**: Image alignment and intensity adjustment of the 3D display are completely automatic using a camera in the loop.

The rest of this paper will describe the system, relate it to previous work and discuss some of the architectural trade-offs. We will then discuss the general issues of multi-view video coding for end-to-end 3D TV, speculating on some multi-view coding approaches that we believe to have the most potential for high compression efficiency.

2. SYSTEM OVERVIEW

Figure 1 shows a schematic overview of our end-to-end 3D TV system. The *acquisition* stage consists of an array of hardware-synchronized cameras. Small clusters of cameras are connected to *producer* PCs. All PCs in our prototype have 3 GHz Pentium 4 processors, 2 GB of RAM, and run Windows XP. The producers capture live, uncompressed video streams and encode them. The coded video streams are then broadcast on separate channels over a *transmission* network, which could be digital cable, satellite TV, or the Internet. In our current system, the producers and consumer PCs are directly connected by gigabit ethernet. This essentially corresponds to a broadband network with infinite bandwidth and almost zero delay.

On the receiver side, individual video streams are decompressed by *decoders*. The decoders are connected by network (e.g., gigabit ethernet) to a cluster of *consumer* PCs. The consumers render the appropriate views and send them to a multiview 3D display. A dedicated *controller* PC broadcasts view parameters to decoders and consumers. The controller is connected to a camera placed in the viewing area for automatic display calibration.

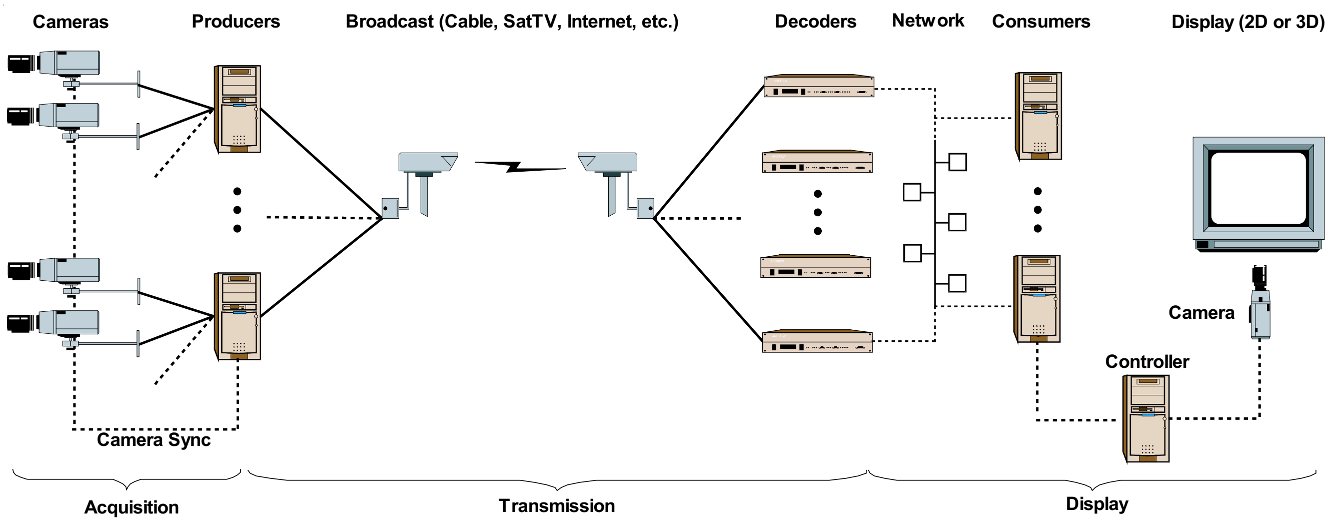


Fig. 1. *A scalable end-to-end 3D TV system.*

2.1. Acquisition

Real-time acquisition of multi-view video has only recently become feasible. Some systems use a 2D array of lenslets or optical fibers in combination with a Fresnel lens in front of a high-definition camera to capture multiple views simultaneously [1]. These systems capture multiple views per lenslet in both horizontal and vertical directions. However, the limited resolution of the camera sensor (fixed at HDTV resolution) allows only a very limited number of lenslets; typically a few hundred in x and y, respectively.

To acquire high-resolution multi-view video requires an array of synchronized cameras. Typically, the cameras are connected to a cluster of PCs [2, 3]. The Stanford multi-camera array [4] consists of up to 128 cameras. Our system uses 16 high-resolution (1300×1030) cameras that capture progressive video at 12 frames per second. Pairs of cameras are connected by IEEE-1394 bus to one of eight producer PCs. A custom-built PCI card generates the synchronization signal for all cameras.

In general, the cameras can be arranged arbitrarily. To cover a large view volume at reasonable cost one can use a number of sparsely arranged cameras [1]. However, to guarantee good image quality on the display side requires to interpolate dense virtual views from the sparse video data. This is generally impossible without a scene model such as per-pixel depth maps [5, 6], or a prior model of the acquired objects. Real-time acquisition of these scene models for general, real-world scenes is very difficult and subject of ongoing research.

Instead, we use a densely-spaced linear array of 16 cameras. The optical axis of each camera is roughly perpendicular to a common camera plane. The advantage of dense camera spacing is that – ideally – the output views on the display correspond to the acquired views of the cameras. In practice, it is impossible to align multiple cameras precisely, and we are using lightfield rendering on the display side to synthesize new views (see Section 2.2). In general, view synthesis for dense camera arrays does not require a scene model, but we could acquire [2] or compute [7, 6] per-pixel depth maps to improve the view-interpolation quality.

2.2. 3D Display

Figure 2 shows a classification of current 3D display technologies. For 3D TV applications, we are only interested in auto-stereoscopic displays. Most of these displays are based on parallax barrier or lenticular technology, and most of them provide multiple stereoscopic images from a wide range of viewing positions. In the following, we examine both parallax and lenticular displays, then present our current 3D display implementation based on multiple projectors.

2.2.1. Parallax and Lenticular Displays

In 1903, F. Ives used a plate with vertical slits as a barrier over an image with alternating strips of left-eye/right-eye images [8]. To extend the viewing angle and viewing position, Kanolt [9] and H. Ives [10] used multiple alternating image stripes per slit (see Figure 3 top). Today's commercial parallax barrier displays use the same idea and place parallax barriers on top of LCD or plasma screens. Parallax barriers generally reduce some of the brightness and sharpness of the image. Some implementations use an LCD screen to display the parallax barriers on top of the viewing screen, which has the advantage that the display can be switched to 2D viewing without any loss in brightness.

Researchers in the 1930s introduced the lenticular sheet, a linear array of narrow cylindrical lenses. Each lens (or

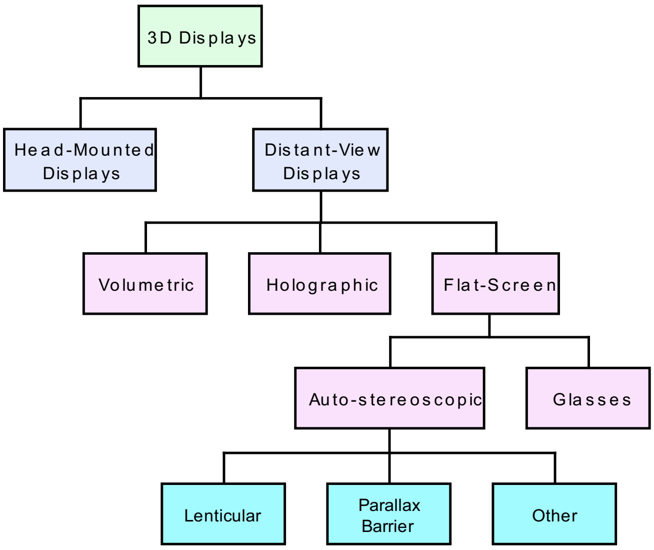
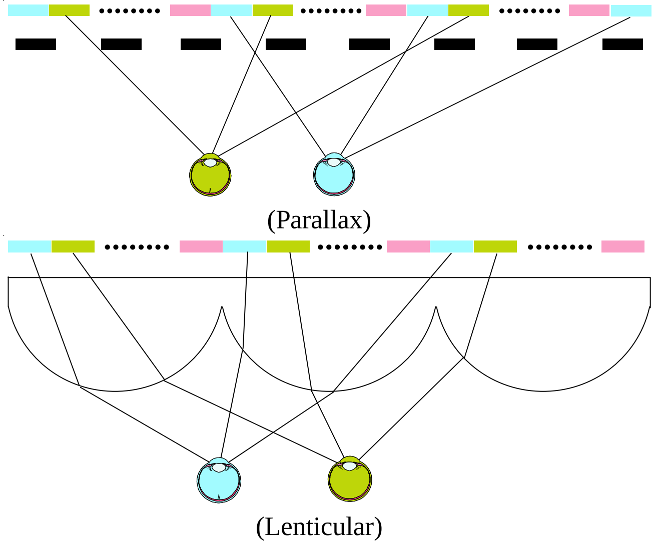


Fig. 2. *Overview of 3D display technologies.*


(Parallax)

(Lenticular)

Fig. 3. *Comparison of Parallax and Lenticular displays.*

lenticule) acts as a light multiplexer, projecting a subset of vertical display pixels towards each eye (see Figure 3 bot). Lenticular images found widespread use for advertising, CD covers, and postcards [11], which has lead to improved manufacturing processes and the availability of large, high-quality, and very inexpensive lenticular sheets. Some modern lenticular displays place diagonally-arranged lenticules on top of high-resolution LCD or plasma monitors.

Most technologies, including parallax-barrier and lenticular methods, provide only horizontal parallax. In 1908, Lippmann proposed using an array of spherical lenses instead of slits [12]. This is frequently called a "fly's-eye" or integral lens sheet, and it provides both horizontal and vertical parallax. Similar to other techniques, integral lens sheets can be put on top of high-resolution LCDs [1, 13]. However, integral displays sacrifice significant spatial resolution in both dimensions to gain full parallax.

2.2.2. Multi-Projector Implementation

A pre-requisite for all auto-stereoscopic displays is that the underlying image has a very high resolution. For example, to be able to display 16 views at HDTV resolution requires $16 \times 1280 \times 720$ or more than 14 million pixels. For SXGA resolution the number goes up to $16 \times 1280 \times 1024$ or more than twenty million pixels. The highest resolution flat-panel screen available today is the IBM T221 LCD with about 9 million pixels.

To be able to display 16 views at XGA (1024×768) resolution today, we implemented both rear-projection and front-projection 3D display prototypes with a linear array of 16 projectors and lenticular screens (see Figure 4). For the rear-projection system (Figure 4 left), two lenticular sheets are mounted back-to-back with optical diffuser material in the center. The front-projection system (Figure 4 right) uses only one lenticular sheet with a retro-reflective front-projection screen material mounted on the back. We tried to match the horizontal separations between cameras and projectors approximately, which required mounting the projectors in separate rows. Two projectors each are connected to one of eight consumer PCs. The large physical dimension $(6' \times 4')$ of our display lead to a very immersive 3D experience.

The two key parameters of lenticular sheets are the field-of-view (FOV) and the number of lenticules per inch (LPI). We use $72'' \times 48''$ lenticular sheets with 30 degrees FOV and 15 LPI. This leads to $180/30 = 6$ viewing zones. At the border between two neighboring viewing zones there is an abrupt view-image change (or "reset") from view number 16 to view number one. This is a fundamental problem for all lenticular or parallax-barrier displays. The only remedy is to increase the number of views and FOV of the display.

Precise manual alignment of the projector array is tedious and becomes downright impossible for more than a handful of projectors or non-planar screens. We use a camera in the loop to automatically compute relative projector poses for automatic alignment [14]. The largest common display area is computed by fitting the largest rectangle of a given aspect ratio (e.g., 4:3) into the intersection of all projected images. Different projectors project images of vastly different intensities. Even worse, the intensity varies quite dramatically over the lifespan of the projector lamp. Using the camera, we determine the minimum intensity per pixel for all projectors and use this information for intensity equalization [15].

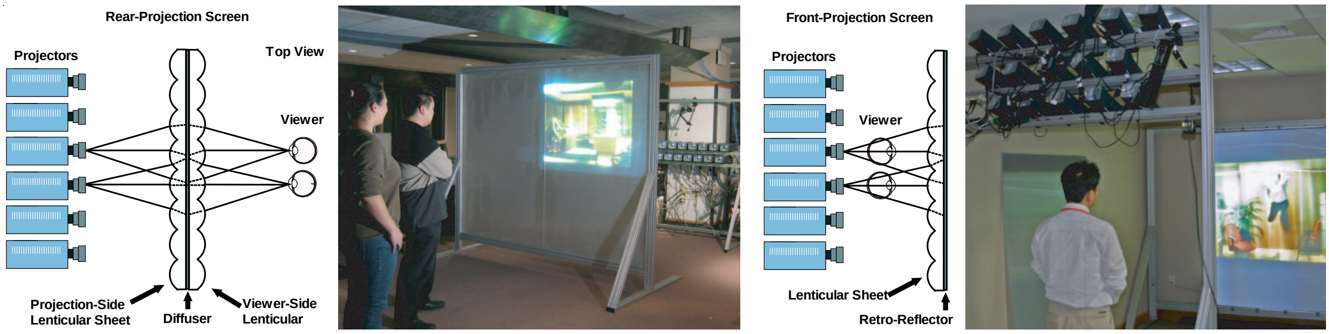


Fig. 4. *Projection-type lenticular 3D displays.*

2.3. View Interpolation

One possible implementation of our 3D TV system uses a one-to-one mapping of cameras to projectors. That means that each video stream is projected by the corresponding projector without view-interpolation. This approach is very simple and scales well, but the one-to-one mapping is not very flexible. For example, the cameras and projectors need to be equally spaced, which is hard to achieve in practice. Moreover, this method cannot handle the case when the number of cameras and projectors is not the same. It also does not provide the ability to interactively control the viewpoint, a feature that has been termed free-viewpoint video.

We have implemented a more flexible approach and use lightfield rendering to synthesize views at the correct virtual camera positions. The display controller requests virtual views by specifying the parameters of virtual cameras. The consumer PCs interpolate new views from the incoming video streams using unstructured lumigraph rendering [16]. The geometric proxy for the rendering is a single plane that can be set arbitrarily. If a per-pixel depth map is available it can also be used during view interpolation.

The performance of our lightfield rendering implementation is completely independent of the total number of transmitted views. Each virtual output view requires only a small number of source frames (e.g., three). In this way, the maximum bandwidth on the network between consumers is limited, which is important to provide scalability. Details of our distributed lightfield rendering can be found in [15].

3. MULTI-VIEW VIDEO CODING

3D TV broadcasting requires that all the views are transmitted to multiple users simultaneously. Transmitting 16 uncompressed video streams with 1280×720 resolution (4:2:0 format) at 30 frames per second requires 5.3 Gb/sec bandwidth, which is well beyond current broadcast capabilities. Clearly, efficient multi-view video coding is important to make 3D TV attractive to broadcasters and network operators. An overview of some early off-line compression approaches that focus on interactive decoding and display applications can be found in [1].

The most straightforward approach to the multi-view coding problem is to temporally encode the individual video streams independent of one another and simulcast each of the views. This simulcast approach is used in our current prototype system, where video streams at full camera resolution (1300×1030) are encoded in real-time with MPEG-2 and decoded on the producer PCs. One major advantage of this approach is that existing video coding standards with commercially available codecs could be used. This allows immediate real-world 3D TV experiments and market studies. On the other hand, the coding efficiency is generally lower than with other multi-view coding approaches.

Another approach to multi-view video compression, promoted by the European ATTEST project [5], is to reduce the data to a single view with per-pixel depth map. This data can be compressed in real-time and broadcast as an MPEG-2 enhancement layer. On the receiver side, stereo or multi-view images are generated using image-based rendering. However, it may be difficult to generate high-quality output because of occlusions or high disparity in the scene. Moreover, a single view cannot capture view-dependent appearance effects, such as reflections and specular highlights.

MPEG has also taken up an interest in this area and for the past couple years has hosted an ad-hoc group to discuss the various requirements and technical issues [17]. Most recently, the group has been working to finalize test conditions that compare proposed multi-view coding approaches to the simulcast approach using the latest H.264/AVC standard. In preparing for these tests, several contributions have been presented to the committee that show gains compared to this benchmark [18, 19, 20, 21]. Thus far, all of the proposed approaches support predictions in both the temporal and spatial dimensions, i.e., over time and across views. In [18], experimental results are reported on a very dense set of multiple view video using multi-direction prediction. In [19], different prediction structures for the picture frames across space and time are investigated; the impact on camera distance and noise in the video on the coding efficiency was also explored. Fecker and Kaup studied the effect of transposing the order of pictures along the spatial direction and using the multiple reference frame prediction of H.264/AVC to achieve predictions in both spatial and temporal dimensions [20]. In [21], an approach that encodes base and intermediate views differently is proposed, where base views utilize only temporal prediction and intermediate views utilize both spatial and temporal predictions; similar approaches have also been presented in [6, 22]. Although the precise manner in which views should be predicted to yield the highest compression is not yet clear, it is evident that a combination of temporal and spatial encoding does have significant potential to provide good results. Some aspects to consider further are required memory and delay, as well as random access.

Although the high degree of coherence among views is expected to yield improved coding results, there are some additional issues to consider to achieve optimal compression efficiency. For one, the cameras are not expected to be perfectly aligned. As a result, some means of rectification that aligns images with respect to the epipolar geometry would be useful in minimizing the residual errors resulting from spatial predictions [23]. Similarly, it is likely that the illumination and color between views is not consistent due to intrinsic parameters of each camera, therefore compensation of illumination and color is also quite important [24].

One final point we would like to make on the compression of multi-view video is with regards to sampling theory - specifically, the minimal number of views required for transmission and rendering. In [25], Lin and Shum derive a lower bound for the minimum number of samples required for lightfield/lumigraph rendering that is closely related to the camera resolution and the scene depth complexity. We believe that the adaptive dropping of segments (e.g., entire frames or blocks) in the multi-view video, which are then interpolated or synthesized at the receiving end, have strong potential to improve the coding efficiency in the rate-distortion sense.

4. RESULTS

Figure 5 shows four images that were taken at different positions on the viewer side of the front-projection display (top) and the corresponding images of the camera array (bot). The parallax of the box in the foreground, the file cabinet on the right, and the door in the background are especially noticeable. The color reproduction between the images of the displayed scene and the actual scene is quite similar. The blur on the 3D display is quite prominent. This is due to the crosstalk between subpixels of different projectors and the light diffusion in the substrate of the lenticular sheets.

The feedback from early users of the system has been mostly positive. We found that dynamic scenes – such as bouncing balls or jumps – are most fun to watch, especially in combination with the freeze-frame feature. It is notable that image quality problems distract from the 3D experience. Most viewers are not willing to give up on the display quality of 2D TV. However, many of the remaining quality problems can be addressed with new high-resolution display technologies, such as organic LEDs or nanotube field-emission displays (FEDs).

5. CONCLUDING REMARKS

We have implemented the first real-time end-to-end 3D TV system with enough views and resolution to provide a truly immersive 3D experience. Additionally, some of the recent work on multi-view coding has been reviewed and the most promising techniques have been discussed.

6. REFERENCES

[1] B. Javidi and F. Okano, Eds., *Three-Dimensional Television, Video, and Display Technologies*, Springer-Verlag, 2002.

[2] T. Naemura, J. Tago, and H. Harashima, “Real-time video-based modeling and rendering of 3D scenes,” *IEEE Computer Graphics and Applications*, pp. 66–73, Mar. 2002.

[3] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, “A real-time distributed light field camera,” in *Proceedings of the 13th Eurographics Workshop on Rendering*. 2002, pp. 77–86, Eurographics Association.

[4] B. Wilburn, M. Smulski, H. K. Lee, and M. Horowitz, “The light field video camera,” in *Media Processors 2002*, Jan. 2002, vol. 4674 of *SPIE*, pp. 29–36.

[5] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. IJsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, and I. Sexton, “An evolutionary and optimised approach on 3D-TV,” in *Proceedings of International Broadcast Conference*, Amsterdam, NL, Sept. 2002, pp. 357–365.

[6] L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 598–606, Aug. 2004.

[7] H. Schirmacher, L. Ming, and H.-P. Seidel, “On-the-fly processing of generalized lumigraphs,” in *Proceedings of Eurographics 2001*. Eurographics Association, 2001, vol. 20 of *Computer Graphics Forum*, pp. 165–173.

[8] F. E. Ives, “Parallax stereogram and process for making same,” U.S. Patent No. 725,567, Apr. 1903.

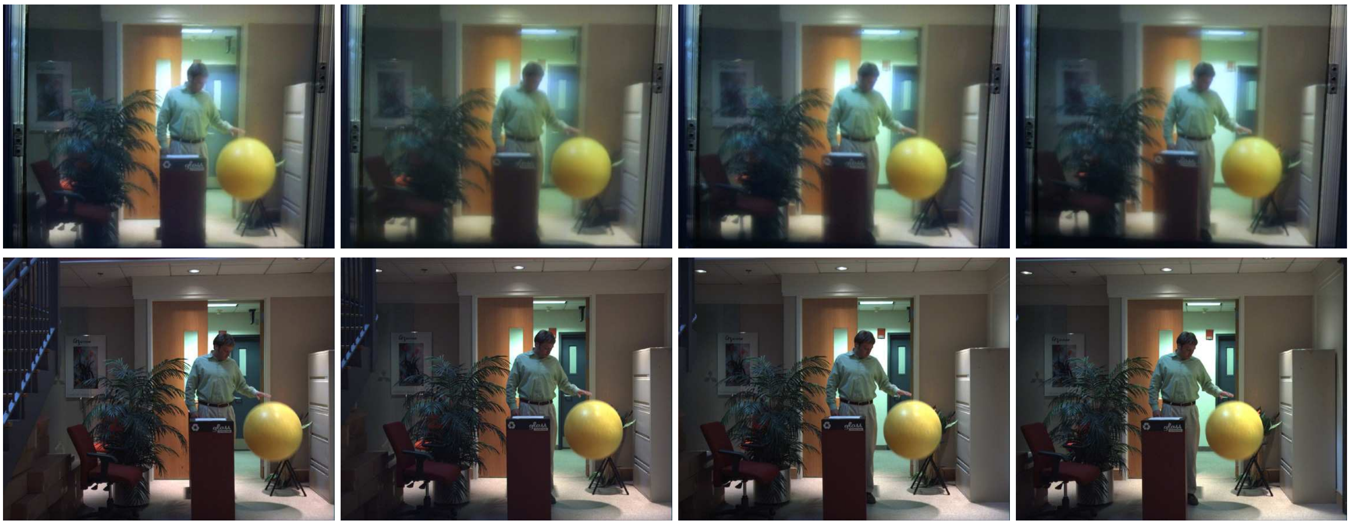


Fig. 5. *Images of a scene from the viewer side of the display (top row) and as seen from some of the cameras (bottom row).*

[9] C. W. Kanolt, “Photographic method and apparatus,” U.S. Patent No. 1,260,682, Mar. 1918.

[10] H. E. Ives, “A camera for making parallax panoramagrams,” *Journal of the Optical Society of America*, , no. 17, pp. 435–439, Dec. 1928.

[11] T. Okoshi, *Three-Dimensional Imaging Techniques*, Academic Press, 1976.

[12] G. Lippmann, “Epreuves reversibles donnant la sensation du relief,” *Journal of Physics*, vol. 7, no. 4, pp. 821–825, Nov. 1908.

[13] S. Nakajima, K. Nakamura, K. Masamune, I. Sakuma, and T. Dohi, “Three-dimensional medical imaging display with computer-generated integral photography,” *Computerized Medical Imaging and Graphics*, vol. 25, no. 3, pp. 235–241, 2001.

[14] R. Raskar, et al., “Multi-projector displays using camera-based registration,” in *IEEE Visualization*, San Francisco, CA, Oct. 1999, pp. 161–168.

[15] W. Matusik and H. Pfister, “3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes,” *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 811–821, Aug. 2004.

[16] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, “Unstructured lumigraph rendering,” in *Computer Graphics*, Los Angeles, CA, 2001, SIGGRAPH 2001 Proceedings, pp. 425–432.

[17] A. Smolic and H. Kimata, “Report on 3DAV exploration,” ISO/IEC JTC1/SC29/WG11 Doc N5878, July 2003.

[18] M. Tanimoto and T. Fuji, “Comparison of temporal and spatial predictions for dynamic ray-space coding,” ISO/IEC JTC1/SC29/WG11 Doc M10668, Mar. 2004.

[19] H. Wang J. Lopez G. Chen, N.-M. Cheung and A. Ortega, “Using inter-view prediction for multi-view video compression,” ISO/IEC JTC1/SC29/WG11 Doc M10512, Mar. 2004.

[20] U. Fecker and A. Kaup, “Transposed picture ordering for dynamic light field coding,” ISO/IEC JTC1/SC29/WG11 Doc M10929, July 2004.

[21] H. Kimata and M. Kitahara, “Preliminary results on multiple view video coding,” ISO/IEC JTC1/SC29/WG11 Doc M10976, July 2004.

[22] Z. F. Gan K. L. Chan S. C. Chan, K. T. Ng and H.-Y. Shum, “The data compression of simplified dynamic light fields,” in *Proceedings of IEEE Int’l Conf. on Acoustics, Speech, and Signal Processing*, Hong Kong, Apr. 2003.

[23] M. Tanimoto and T. Fuji, “Utilization of inter-view correlation for multiple view video coding,” ISO/IEC JTC1/SC29/WG11 Doc M11014, July 2004.

[24] J. Lopez G. Chen, J.H. Kim and A. Ortega, “Illumination compensation for multi-view video compression,” ISO/IEC JTC1/SC29/WG11 Doc M11132, July 2004.

[25] Z. Lin and H.-Y. Shum, “On the number of samples needed in light field rendering with constant-depth assumption,” in *Proceedings of IEEE Conf. Computer Vision Pattern Recognition*, Hilton Head, SC, June 2000, pp. 588–597.