A First Look at a Telepresence System with Room-Sized Real-Time 3D Capture and Life-Sized Tracked Display Wall

Andrew Maimone* Henry Fuchs†

Department of Computer Science
University of North Carolina at Chapel Hill

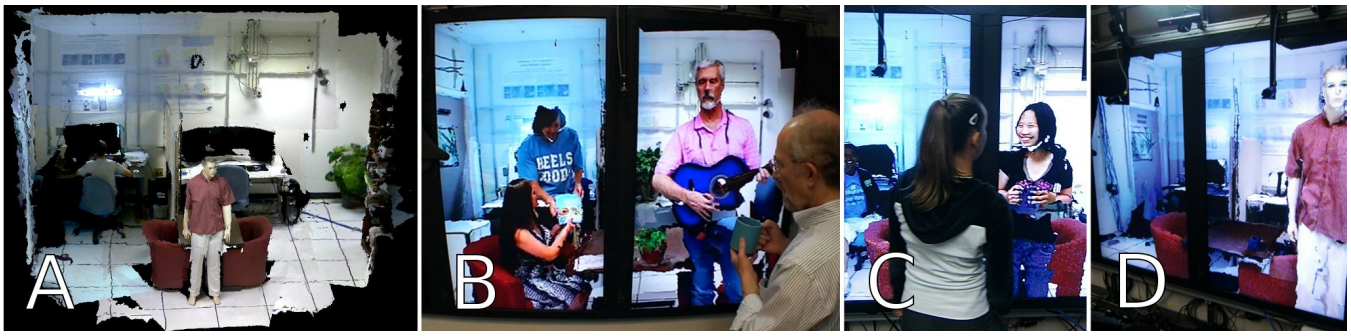

Figure 1: Left to Right: A) System Kinect Coverage. B-C) Users collaborating with remote participants. D) View from far right side.

ABSTRACT

This paper provides a first look at a telepresence system offering room-sized, fully dynamic real-time 3D scene capture and continuous-viewpoint head-tracked display on a life-sized tiled display wall. The system is an expansion of a previous system, based an array of commodity depth sensors. We describe adjustments and improvements made to camera calibration, sensor data processing, data merger, rendering, and display, as required to scale the earlier system to room-sized.

Keywords: teleconferencing, virtual reality, sensor fusion, camera calibration, color calibration, filtering, tracking

Index Terms: H.4.3 [Information Systems Applications]: Communications Applications—Computer conferencing, teleconferencing, and videoconferencing

1 INTRODUCTION

The unification of two remote workspaces through a shared virtual window, allowing remote participants to see each other's environment as a continuation of their own, has been a long-standing goal of telepresence [3, 7].

A recent system by the authors [4] progressed toward this goal by providing fully dynamic 3D scene capture and continuous-viewpoint head tracked 3D display, but the impression that the remote environment was an extension of the viewer's own was limited by the relatively small capture volume (a small office cubicle) and display area (0.43 m^2).

Previous capture systems have demonstrated real-time acquisition of larger volumes (the size of a small room) with various compromises. A 2002 UNC/UPenn system [10] presented an office sized volume, but only the remote collaborator was dynamic; the rest of the scene was a scanned static 3D model. A more recent system by Petit et al. [6] also captures only the remote collaborator, but utilizes a multi-camera setup to offer a larger capture volume. Systems based on interpolation between densely placed 2D cameras, such as the 2004 MERL 3DTV system [5] and the 2010 Holografika system [1] also offer larger capture volumes but do not support continuous viewpoints or vertical parallax.

These limitations of 2D camera systems can be eliminated by providing depth estimates, as in the proposed 3DPresence [8] and Extended Window Metaphor [11] systems and in the demonstrated free-viewpoint television systems of Nagoya University [9], but to our knowledge these systems have not yet demonstrated real-time capture at room scale.

Our new display system, a pair of large (65") conventional 2D display panels with user tracking, is a temporary compromise. It has high resolution (4 MP) and supports continuous viewpoints through encumbrance-free tracking but does not provide a stereo image nor support for multiple tracked users. In comparison, a state-of-the art 3D display, the Holovizio [1], alleviates these issues, but introduces others – lower resolution, a limited field of view and minimum user distance, and lack of vertical parallax. The Holovizio also comes at a much higher cost and complexity.

In this paper, we present an updated telepresence system that supports fully dynamic capture of a small room ($\sim 15\ m^2$) that can be rendered from any of the continuous viewpoints of a tracked user. We believe our system to be the first to incorporate these characteristics at room scale. Furthermore, we have fitted our system with a large tracked display wall that allows the user to become more immersed in the remote scene.

2 BACKGROUND AND CONTRIBUTIONS

The system described in this paper is an extension of earlier work [4] based on an array of Microsoft Kinect™ sensors, widely available, inexpensive ($150) devices that provide matched color images and depth maps. Multiple Kinect sensors were strategically placed and calibrated to provide a unified mesh of the 3D scene which is rendered from the perspective of a tracked user.

*e-mail: maimone@cs.unc.edu
†e-mail: fuchs@cs.unc.edu

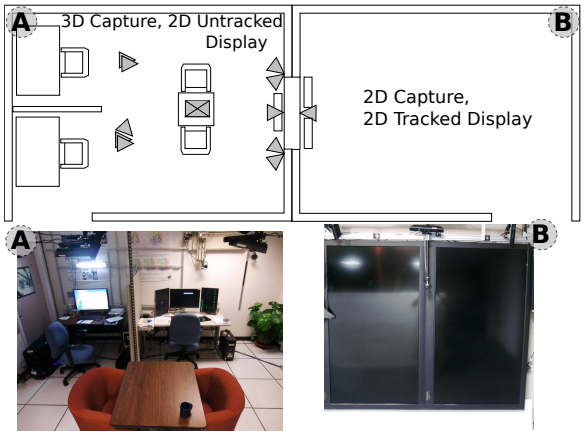

Figure 2: Layout of demonstrated system. Top: Layout of capture and display rooms showing virtual shared wall area. Bottom: Photos of actual capture and display rooms.

To support a much larger capture volume than our previous system [4], the following improvements were made to the system:

1. Calibration procedures were improved to reduce Kinect misalignment, which was more evident in our new larger configuration.

2. Depth data processing was enhanced to reduce the effects of noise caused by increased distances between surfaces and Kinects and by the interference that occurs when multiple Kinects have overlapping views.

3. The software system was enhanced to allow Kinects with a view of only static surfaces (upper walls, ceilings, etc) to be turned off or physically removed, increasing performance and providing more coverage than the total number of physical Kinects allow.

3 System Overview

3.1 Physical Layout

Figure 2 shows the layout of our system. Room A, which offers 3D capture and 2D untracked display of room B, measures 4.3 m $\times$ 4.7 m $\times$ 2.4 m with approximately 75% of the total floor area ($\sim$ 15 m^2) in the capture zone. Room B features 2D capture and a head-tracked perspective 2D display of room A. The two rooms are physically separated, but a view of room A can be seen "through" the display of room B as if the spaces were aligned with a shared hole in the wall (see Figure 2, top). This configuration allows us to demonstrate 3D capture and tracked 2D display while requiring only one set of Kinects and tracked displays.

Figure 4 shows our "ideal" configuration – 3D capture and multi-user autostereoscopic displays are supported in both rooms (C,D).

3.2 Hardware Configuration

Both rooms in our proof-of-concept system share a single PC with a quad-core CPU and a Nvidia GeForce GTX 295 graphics board. Eleven Microsoft Kinect sensors are connected to the PC. The 2D display wall consists of two 1080p 65" LCD panels. We avoided networking and audio in this version of our system since both rooms are served by a single PC and are in close proximity. We plan to address these omissions in a future system.

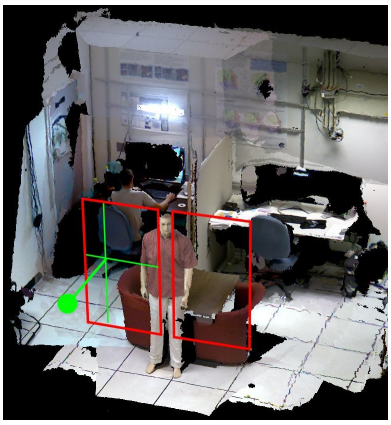

Figure 3: Virtual position of displays (red rectangles) and typical user eye position (green spot) in capture room A of Figure 2.

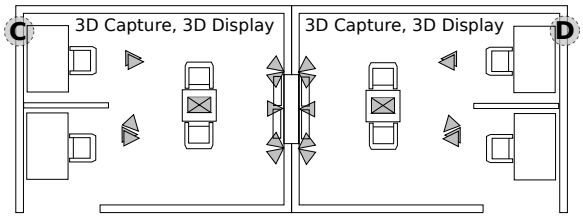

Figure 4: Virtual layout of ideal system

3.3 Software Overview

Rendering We used the same basic rendering pipeline as in our original system [4], but without stereo rendering:

1. When new data is available, read color and depth images from Kinect units and upload to GPU

2. Smooth and fill holes in depth image.

3. For each Kinect's data, form triangle mesh using depth data.

4. For each Kinect's data, apply color texture to triangle mesh and estimate quality at each rendered pixel; render from the tracked user's current position, saving color, quality, and depth values.

5. Merge data for all Kinect units using saved color, quality and depth information.

Tracking We used the same eye tracking method as in our original system [4] – 2D eye detection, depth data, and motion tracking are combined to create a markerless 3D eye position tracker. For the images in this paper and in the supplemental video, the filming camera was tracked in 3D space using a color-coded marker (Figure 5) and the Kinect's depth information.

4 System Enhancements

4.1 Camera Calibration

As previously described [4], we used Zhang's method [13] (as implemented in the OpenCV library) to obtain an initial calibration of the color cameras in the Kinect units using a checkerboard target. However, our new system required several additional considerations:

1. There was no single Kinect unit that shared views with all other units.

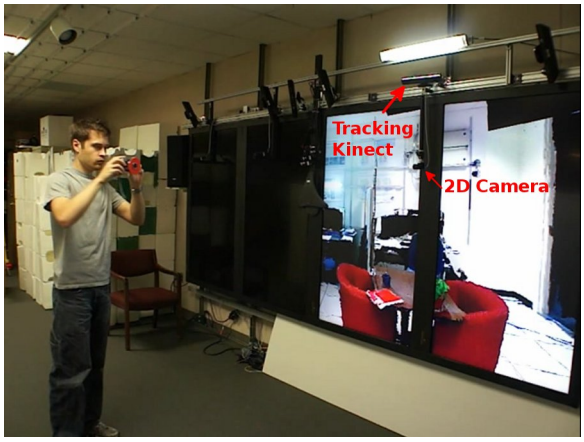

Figure 5: Video camera with colored ring that is detected for tracking, used to provide the tracked images in this paper and in the supplemental video. (Unmarked cameras and deactivated displays visible in image are not used in our system.)

2. Since Kinects are more sparsely placed, the checkerboard target is often located farther from the Kinects, reducing resolution and increasing checkerboard corner detection error.

3. Pairs of Kinects often have overlap at the edges of each other's fields of view, where radial distortion is greatest.

4. Since the user has a greater range of viewing positions, there is more opportunity to look at the scene farther from any one Kinect's line of sight, making depth error more apparent. Since only the pose and distortion parameters of the Kinect color cameras are calibrated, there is no opportunity to correct possible distortions in Kinect depth imagery.

5. Since there are more Kinects than in our previous system, there is a greater opportunity for calibration error to propagate between units.

To address item 1, a camera calibration hierarchy was established that minimized the number of transforms between each Kinect and the reference Kinect. In our demonstrated configuration, at most two transforms were required to transform a Kinect into the reference view.

To address item 2, the Kinect's low framerate/high resolution camera mode was used during calibration and the computed calibration parameters were converted for the low resolution/high framerate mode used during scene capture. To further reduce error resulting from motion blur and from the lack of multi-Kinect synchronization, the checkerboard target was placed at rest before capturing each image.

To address item 3, radial distortion was corrected during scene capture using the distortion coefficients computed during intrinsic calibration of the color camera. Since the the API we are using to communicate with the Kinect, OpenNI[1], automatically registers the depth image to the color image, the same distortion coefficients were applied to both images.

To address items 4 and 5, a supplemental calibration procedure was established. The procedure operates on 3D points as measured by the depth sensor, rather than on 2D projections of points as seen by the color camera, to allow correction of biases in the Kinect's depth readings. The procedure aims to minimize the distance between 3D points measured by all Kinects, reducing the effect of error propagation between Kinects.

[1] http://www.openni.org/

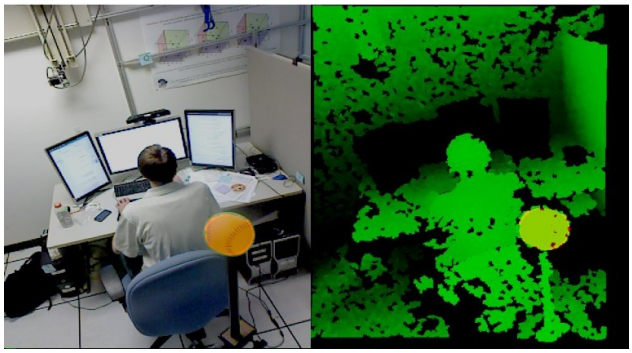

Figure 6: Sphere used for 3D calibration. Left: Green ball detected by its color and highlighted orange in software. Right: Corresponding depth values of ball highlighted orange in depth map.

The procedure is performed follows:

1. An initial calibration is performed using Zhang's method as described above.

2. A spherical object is placed in multiple positions in the capture area. At each position, the sphere is segmented by its color for all Kinects in view and the associated depth values are recorded. This step is illustrated in Figure 6.

3. The depth values from the previous step are fitted to sphere models using RANSAC [2] to eliminate outliers. If the computed radius is too far from true value or if the distances of the sphere center between Kinects are too far apart, the data is rejected.

4. The data for each Kinect is fitted to a affine transform that minimizes the distance between its detected sphere locations and the center of the sphere locations as seen by all other Kinects. RANSAC is used to eliminate outliers. An affine transform fitting was selected over a rigid transform in order to allow linear biases in the Kinect depth imagery to be corrected by scaling.

5. The previous step is repeated until convergence. In practice, between 10 and 100 iterations were performed.

4.2 Depth Data Processing

As previously described [4], interference between Kinect sensors with overlapping views causes holes and additional noise, which can be filled and smoothed in software. These effects are more prominent in our room-sized system as there is more overlap between Kinects – the ones in the rear of the room interfere with others in the rear as well as those in the front. In addition to interference, the Kinects are typically located farther from surfaces in our new system, causing additional depth noise. Another undesirable artifact of our old system was raggedness on edges that represent depth discontinuities. Since the lengths of the ragged edges are related to depth noise, this issue was also more pronounced in our new system.

To reduce the effects of extra noise, the previously described hole filling and smoothing algorithm [4] was enhanced to allow multiple passes of fine scale smoothing (by median filter), which is controlled by parameter N in the revised Algorithm 1. To reduce the appearance of ragged edges, we remove any data that does not meet the previously described hole filling criteria and smoothing criteria [4]; such data occurs at object boundaries along depth discontinuities. The trimming operation is applied to the first t_{trim} passes

of the N passes of the algorithm, allowing control of the degree to which edges are trimmed.

Algorithm 1 Modified N-Pass Median Filter for Hole Filling

```
for pass = 1 to N do
  for i = 1 to numPixels do
    depth_out[i] ← depth_in[i]
    if depth_in[i] = 0 or pass > 1 then
      count ← 0, enclosed ← 0
      v ← {}, n ← neighbors(depth_in[i], radius_pass)
      min ← min(n), max ← max(n)
      for j = 1 to n.length do
        if n[j] ≠ 0 then
          count ← count + 1
          v[count] ← n[j]
          if on_edge(j) then
            enclosed ← enclosed + 1
          end if
        end if
      end for
      if max − min ≤ t_r and count ≥ t_c and enclosed ≥ t_e then
        sort(v)
        depth_out[i] ← v[v.length/2]
      else if pass > 1 and pass <= t_trim then
        depth_out[i] ← 0
      end if
    end if
  end for
  depth_in ← depth_out
end for
```

4.3 Static Kinects

Although we believe that fully dynamic scene capture allows users to better communicate by utilizing surrounding objects, there are often parts of the scene that very rarely change (such as the upper walls and ceiling of a room) but still contribute to the sense of immersion. Providing real-time updates of these static surfaces decreases performance and contributes to interference that occurs between multiple Kinects. Our updated software improves upon this scenario in one of two ways:

1. A Kinect can be temporarily disabled, leaving the last captured frame in the scene. Frame rates increase as the data must no longer be processed and uploaded to the GPU at each frame.

2. A Kinect's last frame can be saved to disk, and the data can be incorporated into future capture sessions even if the Kinect is physically removed. This allows a limited number Kinects to be utilized more effectively. Note that the camera must be returned to its original position and new static data must be captured if the system is recalibrated.

In the latter case, the Kinect data is saved with all other Kinects turned off, eliminating any multi-Kinect interference and improving image quality. In either case, the saved or paused data is rendered just as live data and is color-corrected to match subsequent lighting changes.

We expect that this system could be further improved by automatically and dynamically switching on and off based on movement detected in a scene, trading system performance for latency in the activation of paused Kinects.

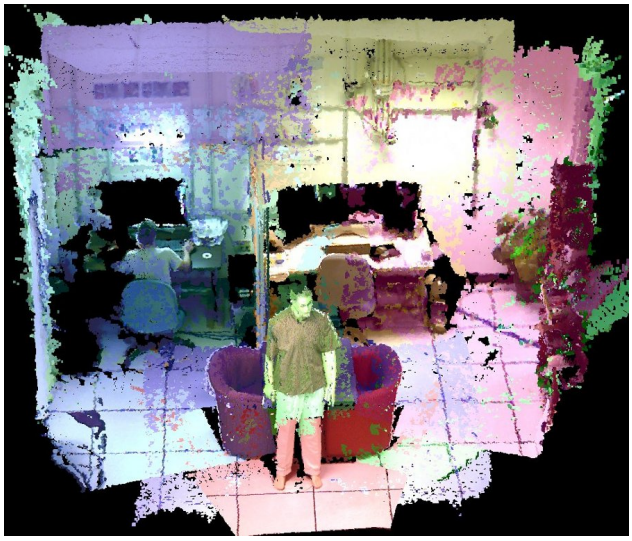

Figure 7: Color-coded coverage of our ten Kinect capture units.

Table 1: 3D Positional Error.

Case	RMS error (cm)
After initial 2D calibration	3.26
Points refitted using 3D affine transform	1.14

4.4 Display

In the new system introduced here, we replaced our autostereoscopic 3D display with a much larger and higher-resolution pair of tiled 2D displays, allowing the remote participant to appear life-sized at a natural interaction distance of 1 m. The tiled display area is approximately 2.5 m^2 (1.76 m x 1.43 m), 8% of which is covered by a 14.6 cm wide bezel. Combined display resolution is 2160×1920 pixels. As before, our display supports only a single head tracked user. In the future, we plan to return to an autostereo 3D display and seek one that can support multiple users and scale to the size of our current 2D display; one such possibility is the Random Hole Display [12].

5 Results

5.1 Kinect Coverage and Calibration Results

Kinect Coverage Figure 1A and Figure 7 show the coverage obtained with the ten Kinect capture units in our updated system. As shown, coverage is provided for most of the surfaces that can be seen by a viewer who is standing near the pictured standing mannequin, facing into the scene.

3D Positional Error 3D Positional error was measured by placing spheres throughout the capture area and measuring the distance between their detected centers between all Kinects with the sphere in view. Figure 8 shows the detected initial locations of each sphere and the locations after refitting using the method described in Section 4.1. These values are quantified in Table 1 – the error values listed are the RMS differences between the detected location of the sphere as seen by each Kinect and the center of the cluster of all Kinects with the sphere in view. In this data set, 98 sphere locations were recorded, seen by an average of 2.23 Kinects each. Each Kinect had between 14 and 40 of the 98 total spheres in view. Figure 9 shows an example of improved calibration using our new methods.

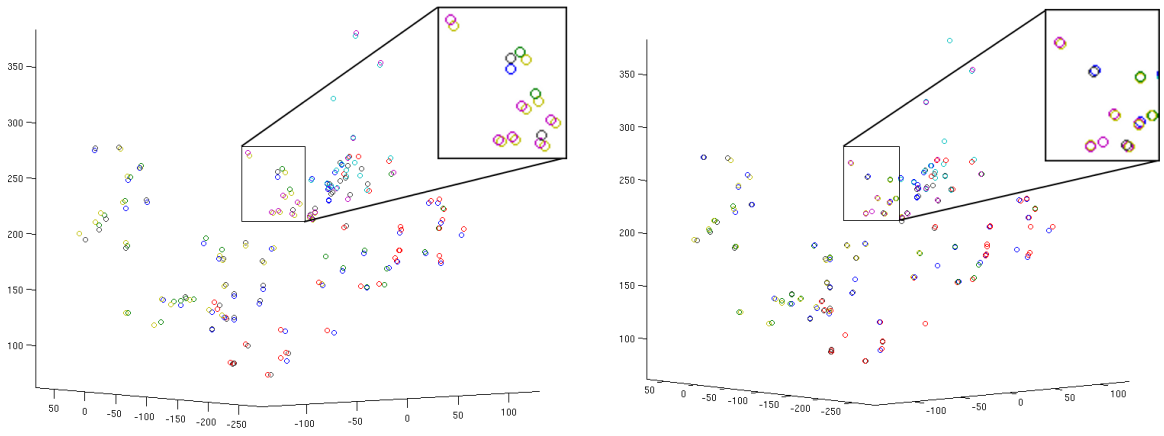

Figure 8: 3D Calibration using detected spheres. Left: Detected locations before adjustment (RMS Error: 3.26 cm). Right: Locations after adjustment (RMS Error 1.14 cm).

Figure 9: Improvement with additional 3D calibration. Left: Initial calibration resulted in misalignment near arm. Right: Arm misalignment improved with additional 3D calibration.

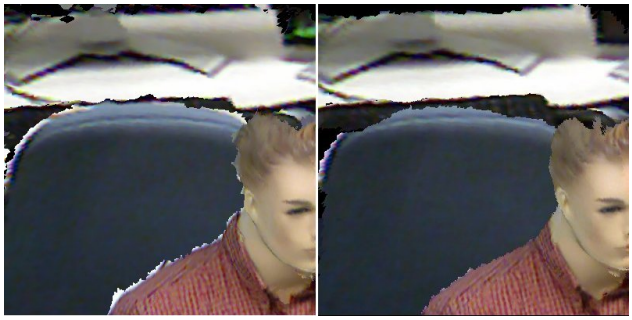

Figure 10: Depth filtering improvement. Left: Depth filtering from [4]. Right: Enhanced depth filtering.

Table 2: Display rates (frames per second)

Rendering Mode	FPS
No enhancements (raw colored point cloud)	62
Meshing, Hole Filling, Data Merger, Tracking	14

5.2 Depth Data Processing Results

Figure 10 shows a comparison of the old and new depth filtering algorithms. In the figure, the revised algorithm removed some of the noisy edges on the mannequin's head and shoulder when set to perform 3 hole filling and smoothing (parameter N) passes and 2 trimming (parameter t_{trim}) passes.

5.3 Display and Tracking

Figures 1B-1D show the system from the perspective of a tracked video camera. Remote users appear life-sized and tracking shows that the view appears correct from several angles, creating a window-like appearance.

5.4 System Performance

Table 2 lists the performance achieved with our test system in two rendering configurations. The system was configured to use data from 7 live capture Kinects, 3 static data Kinects (as described in Section 4.3), and 1 tracking Kinect to render a 2160×1920 view for the display wall. With all enhancements on, display rates remained interactive.

6 Conclusions and Future Work

We have presented solutions to some of the problems related to expanding an earlier telepresence system to room sized: improved calibration techniques, improved data filtering methods, and selectively using live and static data to improve performance.

Using the described methods, we have demonstrated a telepresence system that is able to capture a dynamic, room-sized 3D scene while allowing a remote user to look around the scene from any viewpoint on a life-sized display wall. Using a single PC our system was able to maintain interactive rendering rates.

There are several areas that we would like to improve in our room-size system. Image quality should be further enhanced – images tend to have a noisy look that could be improved with more advanced depth data processing techniques. Our system would also feel more immersive if rendering rates were raised to 30+ Hz.

We also intend to expand our test setup into the "ideal" system shown in Figure 4 by supporting 3D capture and autostereo 3D display for multiple users in both rooms.

ACKNOWLEDGEMENTS

The authors would like to thank Herman Towles, Andrei State, and Jonathan Bidwell for technical discussions and advice, and John Thomas for helping construct some of the camera apparatus. This work was supported in part by the National Science Foundation (award CNS-0751187) and by the BeingThere Centre, a collaboration of UNC Chapel Hill, ETH Zurich, NTU Singapore, and the Media Development Authority of Singapore.

REFERENCES

[1] T. Balogh and P. T. Kovács. Real-time 3d light field transmission. volume 7724, page 772406. SPIE, 2010.

[2] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981.

[3] S. J. Gibbs, C. Arapis, and C. J. Breiteneder. Teleport towards immersive copresence. *Multimedia Systems*, 7:214–221, 1999. 10.1007/s005300050123.

[4] A. Maimone and H. Fuchs. Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, oct. 2011.

[5] W. Matusik and H. Pfister. 3d tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Trans. Graph.*, 23:814–824, August 2004.

[6] B. Petit, T. Dupeux, B. Bossavit, J. Legaux, B. Raffin, E. Melin, J.-S. Franco, I. Assenmacher, and E. Boyer. A 3d data intensive tele-immersive grid. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1315–1318, New York, NY, USA, 2010. ACM.

[7] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 179–188, New York, NY, USA, 1998. ACM.

[8] O. Schreer, I. Feldmann, N. Atzpadin, P. Eisert, P. Kauff, and H. Belt. 3dpresence -a system concept for multi-user and multi-party immersive 3d videoconferencing. In *Visual Media Production (CVMP 2008), 5th European Conference on*, pages 1 –8, nov. 2008.

[9] M. Tanimoto. Overview of free viewpoint television. *Signal Processing: Image Communication*, 21(6):454 – 461, 2006. Special issue on multi-view image processing and its application in image-based rendering.

[10] H. Towles, W.-C. Chen, R. Yang, S.-U. Kum, H. F. N. Kelshikar, J. Mulligan, K. Daniilidis, H. Fuchs, C. C. Hill, N. K. J. Mulligan, L. Holden, B. Zeleznik, A. Sadagic, and J. Lanier. 3d tele-collaboration over internet2. In *International Workshop on Immersive Telepresence, Juan Les Pins*, 2002.

[11] M. Willert, S. Ohl, A. Lehmann, and O. Staadt. The extended window metaphor for large high-resolution displays. In *JVRC10 Joint Virtual Reality Conference of EGVE EuroVR VEC*, pages 69–76. Eurographics, 2010.

[12] G. Ye, A. State, and H. Fuchs. A practical multi-viewer tabletop autostereoscopic display. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 147 –156, oct. 2010.

[13] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666 –673 vol.1, 1999.