

Head-size equalization for improved visual perception in video conferencing

Zicheng Liu, *Senior Member, IEEE*, Michael Cohen, *Senior Member, IEEE*,
Deepti Bhatnagar, Ross Cutler, and Zhengyou Zhang, *Fellow, IEEE*

Abstract—In a video conferencing setting, people often use an elongated meeting table with the major axis along the camera direction. A standard wide angle perspective image of this setting creates significant foreshortening, thus the people sitting at the far end of the table appear very small relative to those nearer the camera. This has two consequences. First, it is difficult for the remote participants to see the faces of those at the far end, thus affecting the experience of the video conferencing. Second, it is a waste of the screen space and network bandwidth because most of the pixels are used on the background instead of on the faces of the meeting participants. In this paper, we present a novel technique, called Spatially-Varying-Uniform scaling functions, to warp the images to equalize the head sizes of the meeting participants without causing undue distortion. This technique works for both the 180 degree views where the camera is placed at one end of the table and the 360 degree views where the camera is placed at the center of the table. We have implemented this algorithm on two types of camera arrays: one with 180 degree view, and the other with 360 degree view. On both hardware devices, image capturing, stitching, and head-size equalization are run in real time. In addition, we have conducted user study showing that people clearly prefer head-size equalized images.

Index Terms—Head-size equalization, video conferencing, visual perception

I. INTRODUCTION

In the past a few years, a lot of progress has been made to improve the audio and video quality during video conferencing. To address the problem of poor audio quality when a speaker is far away from the audio capturing device, researchers have designed microphone arrays and beamforming techniques to capture audio with much higher signal to noise ratio [1], [2], [3]. Intuitively speaking, the effect of beamforming is to virtually pull a far away speaker closer to the audio capturing device.

To address the video capturing problem, people have used pan/tilt/zoom (PTZ) cameras to obtain better images of meeting participants who are far away from the camera. One drawback with PTZ cameras is that they have limited field of view. If they zoom in too much, the context of the meeting room is lost. If they zoom out too much, the meeting participants who sit far away from the camera appear very small. Figure 1 shows a cylindrical projection of a meeting room. We can see that the images of the people sitting at the far end of the table are very small compared to the two people at the front. The remote participants would have to switch views in order to see

the people at the far end thus affecting the video conferencing experience. Furthermore, it is a waste of screen space and network bandwidth because most of the pixels are used on the background instead of on the faces of the meeting participants.

More recently, researchers have designed camera arrays which are placed at the center of a meeting table to capture 360 degree panoramic views [1], [4], [5], [6], [7], [8], [9], [10], [11]. When a meeting table is round, such camera array provides good resolution to all the meeting participants. But unfortunately many meeting tables in practice are elongated. For elongated meeting tables, it has the same problem that people's head sizes are not uniform due to the distances to the camera. Figure 2 shows an image captured by an omnidirectional camera placed at the center of a meeting table. The table size is 10×5 feet. We can see that the person in the middle of the image appears very small compared to the other two people because he is further away from the camera.

In this paper, we present a novel technique, called Spatially-Varying-Uniform scaling functions, to warp the images to equalize the head sizes of the meeting participants without causing undue distortion. This is analogous to audio beamforming in that we virtually pull those participants which sit far away from the camera closer.

This algorithm works for both the 180 degree views where the camera is placed at one end of the table and the 360 degree views where the camera is placed at the center of the table. We have implemented this algorithm on two types of hardware devices. The first is a 5 camera array with nearly 180 degree field of view. The five individual cameras in the camera array have different field of views so that the camera at the center has enough resolution to capture people who sit at the far end of the table. This camera is usually placed at one end of the meeting table to capture the entire room. The second is a 5 camera array with 360 degree field of view. This camera is usually placed at the center of the meeting table to capture a 360 degree panoramic view of the meeting room. On both hardware devices, image capturing, stitching, and head-size equalization are run in real time. In addition, we have conducted user study which shows that head-size equalization indeed improves people's perception.

II. SPATIALLY VARYING UNIFORM SCALING FUNCTION

In this section, we describe a parametric class of image warping functions that attempt to equalize people's head sizes in the video conferencing images. We call the class of warping functions *Spatially Varying Uniform Scaling* functions, or

Dr. Zicheng Liu, Dr. Michael Cohen, Dr. Ross Cutler, and Dr. Zhengyou Zhang are with Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-mail: {zliu,mcohen,rcutler,zhang}@microsoft.com

Deepti Bhatnagar is with Indian Institute of Technology, New Delhi, India.



Fig. 1. A cylindrical projection of a meeting room where the camera device is placed at one end of the table.



Fig. 2. A meeting room captured by an omnidirectional camera. The camera device is placed at the center of the table. The table dimension is 10×5 feet.

SVU scaling for short. These functions locally resemble a uniform scaling function to preserve aspect ratios, however, the scale factor varies over the image to create the warp. The class of *conformal* projections can provide local uniform scaling, however, they introduce rotations which are visually disturbing. This led us to the SVU scaling functions that avoid rotations at some costs in terms of introducing shear.

We will use the example shown in Figure 1 to describe the SVU scaling. The images are captured in real-time using a five-lens device we describe later. After stitching, this provides us with a full 180 degree cylindrical projection panoramic image.

We would like the warping function to be such that it zooms up the center more than the sides while locally mimicking a uniform scaling. We would like to avoid rotations (as might appear in conformal projections), particularly keeping vertical lines vertical. The warp we initially describe induces some vertical shear, thus slanting horizontal lines. We describe at the end of this section a modification that corrects for much of this at some cost to aspect ratio near the top and bottom boundaries.

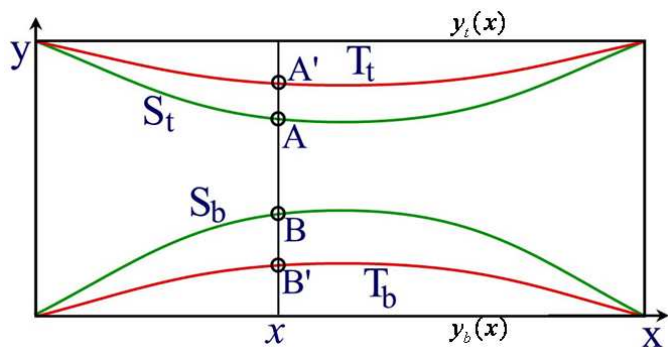


Fig. 3. The warping function is determined by two sets of curves: source (green) and target (red) curves.

The SVU scaling function depends on two curves, represented as piecewise cubic splines, as shown in Figure 3. These two *source* curves define common (real world) horizontal features such as the tops of people's heads, and the edge of the table. The two curves can be either marked by a user manually, or estimated automatically by segmenting the boundary of the meeting table [12]. A factor, α which is a parameter specified by the user determines how much the image is warped.

Let $y = S_t(x)$ and $y = S_b(x)$ be the equations of the top and bottom source curves respectively. Two *target* curves (where points on the source curves will move to) are determined by the source curves and α . If we denote the equation of the line between the end points of $S_t(x)$ as $y = y_t(x)$, and the equation of line connecting the bottom source ends as $y = y_b(x)$, then the top target curve is

$$T_t(x) = (1 - \alpha)S_t(x) + \alpha y_t(x), \quad (1)$$

and the bottom target curve is

$$T_b(x) = (1 - \alpha)S_b(x) + \alpha y_b(x). \quad (2)$$

An $\alpha = 0$ will leave the image untouched. An $\alpha = 1$ will pull pixels on source curves to the lines between the end points. For example, the four curves shown in Figure 3 consist of two green source curves and two red target curves.

Given any vertical scanline x as shown in Figure 3, let A, B denote its intersections with the source curves, and A', B' the intersections with the target curves. The SVU scaling function will scale AB to $A'B'$. Let

$$\begin{aligned} r(x) &= \frac{\|A'B'\|}{\|AB\|} \\ &= \frac{T_t(x) - T_b(x)}{S_t(x) - S_b(x)} \end{aligned} \quad (3)$$

We scale the line vertically by $r(x)$, and to preserve aspect ratio we also scale the scanline horizontally by $r(x)$. Therefore,

the total width of the new image, w' , becomes

$$w' = \int_0^w r(x)dx \quad (4)$$

where w is the width of the source image.

For any pixel (x, y) in the source image, let (x', y') denote its new position in the warped image. We have

$$\begin{aligned} x' &= \int_0^x r(x)dx \\ y' &= T_t(x) + r(x) * (y - S_t(x)) \end{aligned} \quad (5)$$

This is the forward mapping equation for the SVU scaling function. The SVU scaling function is not a perfect uniform scaling everywhere. It is easy to prove that the only function that is a perfect uniform scaling everywhere is a uniform global scaling function.

To avoid hole filling, we use backward mapping in the implementation. Denote

$$R(x) = \int_0^x r(x)dx. \quad (6)$$

The backward mapping equation is

$$\begin{aligned} x &= R^{-1}(x') \\ y &= \frac{y' - T_t(x)}{r(x)} + S_t(x) \end{aligned} \quad (7)$$

A. Horizontal Distortion Correction

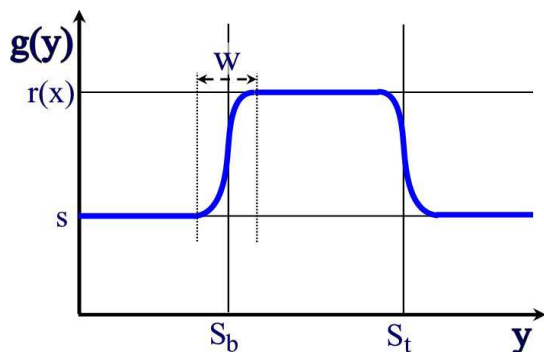


Fig. 4. The vertical scale function.

While the SVU-scaling function maintains vertical lines as vertical, it distorts horizontal lines. The distortions are smallest between the source curves and largest near the top and bottom. Scenes often contain horizontal surfaces near the top or bottom, such as a table and the ceiling on a room for which the distortions may be noticeable (see Figure 1). To minimize this problem we relax the uniformity of the scaling and nonlinearly scale each vertical scanline. The portion of the image between the source curves is scaled by $r(x)$ as described above. The portions outside the source curves are scaled less in the vertical direction. The horizontal scaling remains the same (i.e., $r(x)$) to maintain the straightness of vertical lines. To maintain continuity, the vertical scaling function smoothly transitions as it crosses the source curves.

Consider the vertical line in Figure 3. Denote $g(y)$ to be the vertical scale factor at any point y on this vertical line (see

Figure 4). Note that $g(y)$ is dependent on x . $g(y)$ is controlled by two parameters s and ω . The portion of the vertical scanline more than $\omega/2$ distance from the source curves is scaled by $r(x)$ between the source curves and by s outside the source curves. The three constant segments are glued together by two cubic splines in $[S_t - 0.5\omega, S_t + 0.5\omega]$ and $[S_b - 0.5\omega, S_b + 0.5\omega]$. Each cubic spine has ends with values s and $r(x)$ and a slope of 0 at both ends.

The parameter ω controls the continuity at the source curves. For example, if the scene is discontinuous at the source curves, one can choose a very small ω without noticeable artifacts. In the special case when $s = r(x)$, $g(y)$ becomes a constant which is what we assume in deriving Equation 5.

III. HALF-RING CAMERA ARRAY



Fig. 5. The half-ring camera which consists of five 1394 fire-wire video cameras. The center camera has the smallest field of view.

If we directly apply our warping function, the extreme enlargement of the far people will be very blurry due to the limited resolution of the image in this area. To solve this problem, we have built a special "half-ring" video camera consisting of five inexpensive ($< \$50$ each) fire-wire video cameras daisy-chained together (See Figure 5). A single IEEE 1394 fire-wire delivers five video streams to the computer. The resolution of each camera is 640×480 . Each camera has a different lens. Figure 6 shows the five images directly from the five video cameras. The center camera has the smallest field of view (about 25 degrees) to provide enough resolution for the distance. The field of view of the two cameras next to the center are 45 degrees, with the outer having the largest field of view (60 degrees). Together, they cover 180 degrees with enough overlap between neighboring cameras for calibration and image stitching.

We use well-known techniques to calibrate these cameras and compute the homography between the cameras [13], [14], [15], [16]. We then stitch the individual images together to generate a 180 degree cylindrical image (see Figure 1). Computation overhead is reduced at run time by pre-computing a stitch table that specifies the mapping from each pixel in the cylindrical image to pixels in the five cameras. For each pixel in the cylindrical image, the stitch table stores how many cameras cover this pixel, and the blending weight for each camera. Blending weights are set to one in most of the interior of each image with a rapid fall off to zero near the edges. Weights are composed with an *over* operator where the higher resolution pixel is composed over a lower resolution one. At



Fig. 6. Images captured by the five individual cameras in the half-ring camera array.

run time, we use a look up the table to perform color blending for each pixel.

A. SVU Scaling the Stitch Table

Applying the SVU scaling function to the stitched image would result in a loss of resolution. Instead, we apply the SVU scaling function to the stitch table itself, and generate a concatenated table. During this offline concatenation, we use bilinear interpolation on both the pixel positions and camera weights to fill in zoomed-up regions to avoid losing resolution. Below is a more detailed description.

Assume there are n individual cameras. Let I_c denote the image from camera c where $c = 1, \dots, n$. Let I_s denote the stitched image and I' the image after SVU scaling. For any given pixel location (i, j) in image I' , let $(u(i, j), v(i, j))$ denote its corresponding location in I_s . $(u(i, j), v(i, j))$ is computed by the backward mapping equation (Equation 7).

Note that in general $(u(i, j), v(i, j))$ are not integers. We use bilinear interpolation to obtain the corresponding location in each image I_c . Given any pixel location (x, y) in the stitched image, let $p_c^s(x, y)$ denote the corresponding location in I_c , and $w_c^s(x, y)$ denote the weight of the c th camera for this pixel. Both $p_c^s(x, y)$ and $w_c^s(x, y)$ are stored in the stitch table. Note that $p_c^s(x, y)$ is a floating point number. Given any pixel location (i, j) in image I' , its corresponding location in image I_c is computed by the following bilinear interpolation equation:

$$p'_c(i, j) = \sum_{k=0}^1 \sum_{l=0}^1 \lambda_k(u) \lambda_l(v) p_c^s(\lfloor u \rfloor + k, \lfloor v \rfloor + l) \quad (8)$$

where $\Delta u = u - \lfloor u \rfloor$, $\Delta v = v - \lfloor v \rfloor$, $\lambda_k(u) = k\Delta u + (1 - k)(1 - \Delta u)$, and $\lambda_l(v) = l\Delta v + (1 - l)(1 - \Delta v)$.

Similarly we obtain the bilinear interpolation equation for the weight of the c th camera corresponding to pixel location (i, j) in image I' :

$$w'_c(i, j) = \sum_{k=0}^1 \sum_{l=0}^1 \lambda_k(u) \lambda_l(v) w_c^s(\lfloor u \rfloor + k, \lfloor v \rfloor + l) \quad (9)$$

Both $p'_c(i, j)$ and $w'_c(i, j)$ are stored in the concatenated table. At run time, the intensity of I' at pixel location (i, j) is calculated as

$$I'(i, j) = \sum_{c=1}^n w'_c(i, j) I_c(p'_c(i, j)) \quad (10)$$

Note that the format of the concatenated table is the same as the stitch table, and the operation at each pixel is also the same.

Therefore, SVU scaling does not result in any additional run time overhead. Since each pixel in I' is accessed exactly once, the computational complexity of the combined SVU scaling and stitching is linear in the number of pixels of I' .

IV. APPLICATION TO THE IMAGES CAPTURED BY OMNIDIRECTIONAL CAMERAS

SVU scaling functions can be applied to images captured by omnidirectional cameras as well. For images captured by an omnidirectional camera, the shapes of the two source curves are different due to the camera position. Figure 7 shows the two source curves on the 360 degree panoramic image as shown in Figure 2. As we can see, the bottom curve has two peaks, and the top curve has two valleys. The line connecting the two peaks of the bottom curve is used as $y = y_b(x)$ (See Figure 3), while the line connecting the two valleys of the top curve is used as $y = y_t(x)$. The two target curves can be computed in the same way as in Equations 1 and 2. And the rest of the algorithm is carried out in the same way as in the 180 degree image case. Figure 8 shows the result after applying SVU scaling function to the image in Figure 2.

As the sensor technology rapidly advances, people are designing inexpensive high resolution (over 2000 pixels in horizontal resolution) omnidirectional video cameras [1] for video conferencing. But due to network bandwidth and client's screen space, only a smaller-sized image can be sent to the client. The SVU scaling function provides a much better way to effectively use the pixels to optimize the user's experience. Notice that by concatenating SVU scaling table with the stitch table, the zoomed up pixels will not become blurry because there are enough pixels in the images captured by the individual cameras.

V. RESULTS

A. Results on images captured with the half-ring camera

For the image in Figure 1, Figure 9 shows both the source and target curves with $\alpha = 0.3$. Figures 10 through 13 show the results of using the SVU scaling function. Figure 10 shows the result of applying the SVU scaling function without correcting horizontal distortion. Figure 11 shows the result after correcting for horizontal distortion with $\omega = 0.8$. By comparing Figure 11 with Figure 10, we can see that after horizontal distortion correction, the surface of the meeting table surface becomes flat and so does the ceiling. Finally we show some results with different α . Figure 12 shows the result with $\alpha = 0.2$, and Figure 13 shows the result with $\alpha = 0.4$. We would like to point out that one of the individual cameras



Fig. 7. The two curves on an 360 degree panorama image have different shapes compared with 180 degree images.



Fig. 8. After applying SVU scaling to the image in Figure 2.

(the second image from left in Figure 6) does not focus well because the lens we bought has some defect. As a result, the image captured by this camera is somewhat blurry.

During live meetings, we store multiple tables corresponding to different α 's so that one can change levels in real time. The size of the stitched image is approximately 300 by 1200 pixels. During warping, we keep the image width the same, and as a result, the image height decreases as we zoom up. The frame rate is 10 frames per second on a CPU with a single 1.7GHZ processor. The delay is approximately 65 milliseconds.

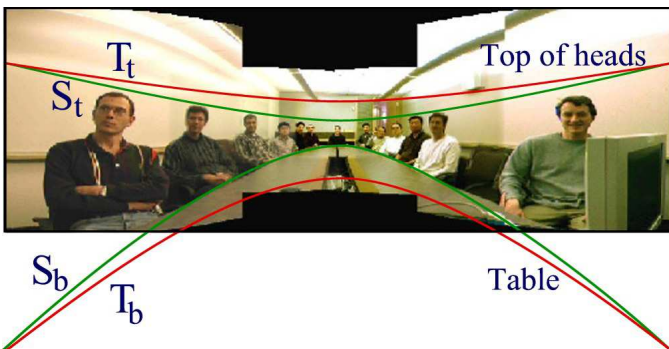


Fig. 9. The source curves and the target curves with $\alpha = 0.3$.

B. Results on images captured with an omnidirectional camera

We have experimented the head-size equalization algorithm with a new prototype of the omnidirectional camera device [1]. We performed experiments in three different meeting room settings. Figure 14, 16, and 18 each shows a sample frame from the three different room settings. For the first setting (Figure 14), the table dimension is 12×5 feet. For the second setting (Figure 16), the table dimension is 16×5 feet. The third setting (Figure 16) is the same as the second setting except that a person stands up and walks around the table.

Figures 15, 17, and 19 are the results after applying SVU scaling function with $\alpha = 0.5$ on the images in Figures 14, 16, and 18, respectively.

VI. USER STUDY

We have conducted user study to check whether head-size equalization improves people's perception. We use the data captured in three different room settings as shown in Figure 14, 16, and 18. Each video is about 30 seconds long. For each room setting, we generate five video sequences. The first video sequence is the original stitched video sequence without SVU scaling. The other four sequences are generated by applying SVU scaling with $\alpha = 0.3, 0.5, 0.7,$ and 0.9 , respectively. For each setting, a user is presented with the five video sequences side-by-side on a 1024×768 computer screen. The reason we chose 1024×768 screen resolution is that this is the typical screen size that a remote meeting participant may use in practice. For each setting, the user is required to rank the five video sequences. A rank of 1 means the best, and a rank of 5 is the worst. In addition, the user is asked to compare his/her best-ranked video sequence with the original stitched video sequence and give an opinion score ranging from 1 to 5 where a score of 3 means the best-ranked video sequence looks the same as the original video sequence, a score of 1 means the best-ranked sequence looks much worse than the original video sequence, and 5 means the best-ranked sequence looks much better.

There are 17 users who participated in the user study. Figure 20 shows the user study ranking results where the horizontal axis is the α value and the vertical axis is the average ranking. The solid curve in Figure 20 is the user study result for the first room setting where the meeting table size is 12×5 feet. The dashed curve is the result for the second setting where the meeting table size is 16×5 feet. The dotted curve is the result for the third setting where a person walks around the room. We can see that for all three settings, the original stitched sequence without SVU-scaling is ranked the worst. For the first setting, $\alpha = 0.5$, $\alpha = 0.7$, and $\alpha = 0.9$ are all ranked very high. It suggests that there are no visible distortions even for large α values, and people prefer the images after head size equalization. For the second and third setting, the meeting table is extremely long thus resulting in visible distortions when α is large. That is why $\alpha = 0.5$ is ranked the best in both settings. The user study result suggests that $\alpha = 0.5$ is a safe choice for all table sizes.



Fig. 10. SVU scaling without horizontal distortion correction.



Fig. 11. SVU scaling with horizontal distortion correction.

Fig. 12. SVU scaling with $\alpha = 0.2$.Fig. 13. SVU scaling with $\alpha = 0.4$.

The average opinion scores for the three room settings are 4.52, 4.42, and 4.41, respectively. We can see that in all three room settings, people strongly prefer the results after head-size equalization.

VII. CONCLUSION

We have presented a technique, called SVU scaling function, for head-size equalization during video conferencing. The algorithm is fast and can be easily combined with panorama stitching operation so that it does not add additional computational overhead. We have applied this technique to 180 degree wide angle images captured by a half-ring camera as well as 360 degree panoramic images captured by an omnidirectional camera. We have conducted user study which shows that people clearly prefer images after head-size equalization.

One limitation of our current system is that the user has to invoke a calibration program to re-compute SVU scaling function when there is a change on the orientation or position of the camera device. We have developed a technique to estimate the relative position and orientation of a camera device

automatically [12], but it is computationally too expensive to run at every frame. In the future, we would like to develop a technique to automatically track the position and orientation of the camera device with very small computational overhead.

REFERENCES

- [1] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L.-W. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg, "Distributed meetings: A meeting capture and broadcasting system," in *ACM Multimedia*, 2002.
- [2] Y. Tamai, S. Kagami, H. Mizoguchi, K. Sakaya, K. Nagashima, and T. Takano, "Circular microphone array for meeting system," *Proceedings of IEEE*, vol. 2, no. 2, pp. 1100–1105, October 2003.
- [3] D. G.-P. H. K. Maganti and I. McCowan, "Speech enhancement and recognition in meetings with an audio-visual sensor array," *Technical Report IDIAP-RR 06-24*, IDIAP Research Institute and Ecole Polytechnique Federale de Lausanne (EPFL), 2006.
- [4] S. Nayar, "Omnidirectional video camera," in *DARPA Image Understanding Workshop*, 1997.
- [5] —, "Catadioptric omnidirectional camera," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1997.
- [6] M. Aggarwal and N. Ahuja, "High dynamic range panoramic imaging," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.



Fig. 14. A sample frame of the video captured in the first room setting. The table dimension is 12×5 feet.



Fig. 15. The result of applying SVU scaling with $\alpha = 0.5$ to the image shown in Figure 14.



Fig. 16. A sample frame of the video captured in the second room setting. The table dimension is 16×5 feet.



Fig. 17. The result of applying SVU scaling with $\alpha = 0.5$ to the image shown in Figure 16.

- [7] S. Coorg, N. Master, and S. Teller, "Acquisition of a large pose-mosaic dataset," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 872–878.
- [8] S. Nayar and A. Karmarkar, "360x360 mosaics," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000, pp. II388–392.
- [9] J. Yang, X. Zhu, R. Gross, J. Kominek, Y. Pan, and A. Waibel, "Multimodal people id for a multimedia meeting browser," in *ACM Multimedia*, Orlando, Florida, 1999, pp. 159–168.
- [10] R. A. Hicks and R. Bajcsy, "Catadioptric sensors that approximate wide-angle perspective projections," in *Workshop on Omnidirectional Vision*, 2000, pp. 97–103.
- [11] M. V. T. K. C. Vallespi, F. D. L. Torre, "Automatic clustering of faces in meetings," in *IEEE International Conference on Image Processing*, Atlanta, Georgia, October 2006.
- [12] Y. Chang, R. Cutler, Z. Liu, Z. Zhang, A. Acero, and M. Turk, "Automatic head-size equalization in panorama images for video conferencing," in *IEEE International Conference on Multimedia and Expo*, Amsterdam, The Netherlands, July 2005.
- [13] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *International Conference on Computer Vision (ICCV'99)*, 1999, pp. 666–673.
- [14] M. Irani, P. Anandan, and S. Hsu, "Mosaic based representations of video sequence and their applications," in *International Conference on Computer Vision (ICCV'95)*, 1995, pp. 605–611.
- [15] S. Mann and R. W. Picard, "Virtual bellows: Constructing high quality images from video," in *First IEEE International Conference on Image Processing (ICIP'94)*, 1994, pp. I:363–367.
- [16] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *Computer Graphics, Annual Conference Series*. Siggraph, 1997, pp. 251–258.

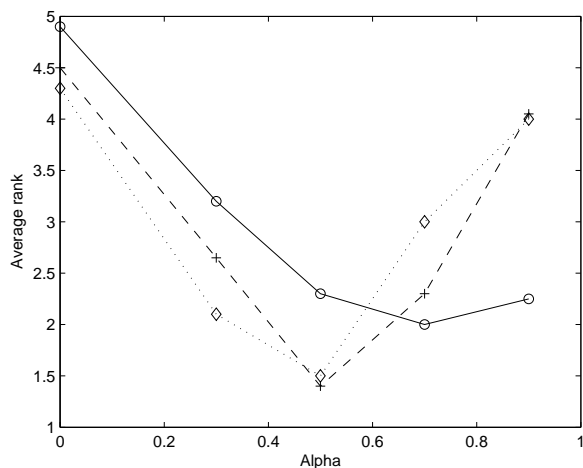


Fig. 20. Results of the user study. Solid curve: user study result for the first room setting where the meeting table size is 12×5 feet. Dashed curve: user study result for the second setting where the meeting table size is 16×5 feet. Dotted curve: user study result for the third setting where a person walks around the room. The horizontal axis is the α value and the vertical axis is the average ranking.



Fig. 18. A sample frame of the video captured in the third room setting where a person walks around the table.



Fig. 19. The result of applying SVU scaling with $\alpha = 0.5$ to the image shown in Figure 18.