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]–[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. Fig. 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 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]–[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. Fig. 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 five-camera array with a 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 five-camera array with a 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 a user study which shows that head-size equalization indeed improves people's perception.

Manuscript received October 25, 2006; revised May 10, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Qian Zhang.

Z. Liu, M. Cohen, R. Cutler, and Z. Zhang are with Microsoft Research, Redmond, WA 98052 (e-mail: zliu@microsoft.com; mcohen@microsoft.com; rcutler@microsoft.com; zhang@microsoft.com).

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

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2007.906571

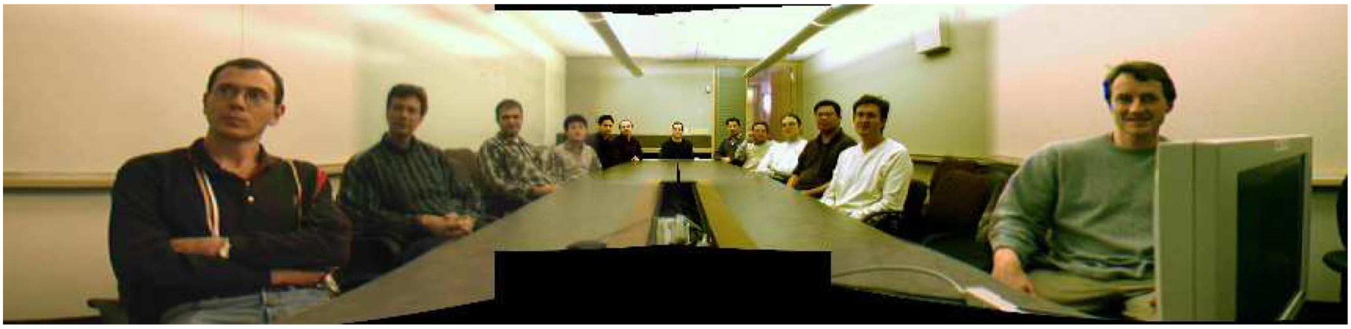


Fig. 1. Cylindrical projection of a meeting room where the camera device is placed at one end of the table.

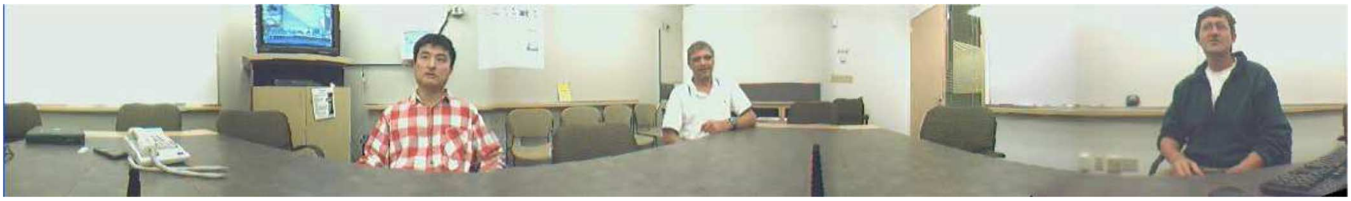


Fig. 2. Meeting room captured by an omnidirectional camera. The camera device is placed at the center of the table. The table dimension is 10 × 5 feet.

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 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 Fig. 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.

The SVU scaling function depends on two curves, represented as piecewise cubic splines, as shown in Fig. 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

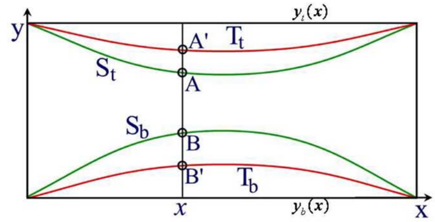


Fig. 3. Warping function is determined by two sets of curves: source (green) and target (red) curves.

$$T_t(x) = (1 - \alpha)S_t(x) + \alpha y_t(x) \tag{1}$$

and the bottom target curve is

$$T_b(x) = (1 - \alpha)S_b(x) + \alpha y_b(x). \tag{2}$$

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 Fig. 3 consist of two green source curves and two red target curves.

Given any vertical scanline x as shown in Fig. 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

$$r(x) = \frac{\|A'B'\|}{\|AB\|} = \frac{T_t(x) - T_b(x)}{S_t(x) - S_b(x)}. \tag{3}$$

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

$$w' = \int_0^w r(x)dx \tag{4}$$

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

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

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

$$R(x) = \int_0^x r(x)dx. \tag{6}$$

The backward mapping equation is

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

A. Horizontal Distortion Correction

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 Fig. 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 Fig. 3. Denote $g(y)$ to be the vertical scale factor at any point y on this vertical line (see Fig. 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 (5).

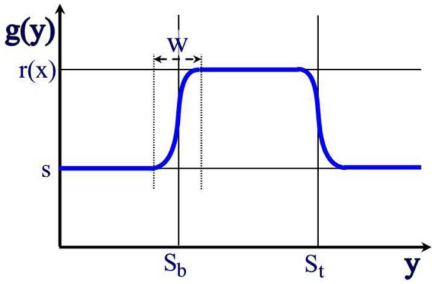


Fig. 4. Vertical scale function.

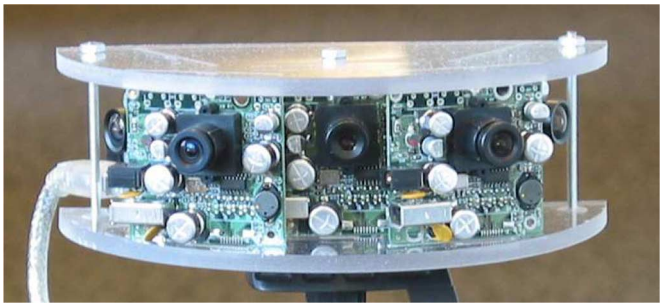


Fig. 5. Half-ring camera which consists of five 1394 fire-wire video cameras. The center camera has the smallest field of view.

III. Half-Ring Camera Array

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 Fig. 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. Fig. 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]–[16]. We then stitch the individual images together to generate a 180-degree cylindrical image (see Fig. 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

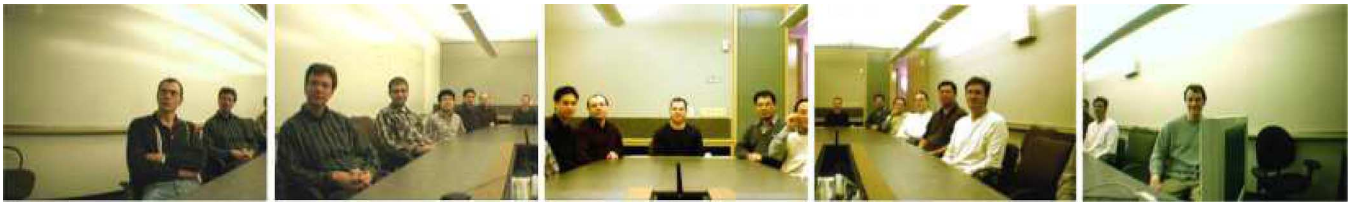


Fig. 6. Images captured by the five individual cameras in the half-ring camera array.

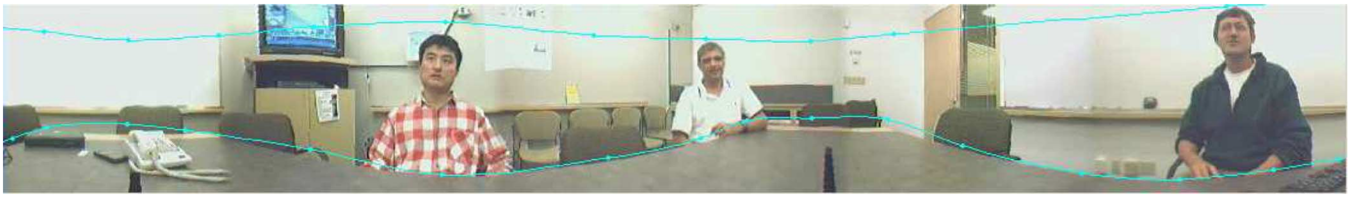


Fig. 7. Two curves on an 360-degree panorama image have different shapes compared with 180-degree images.

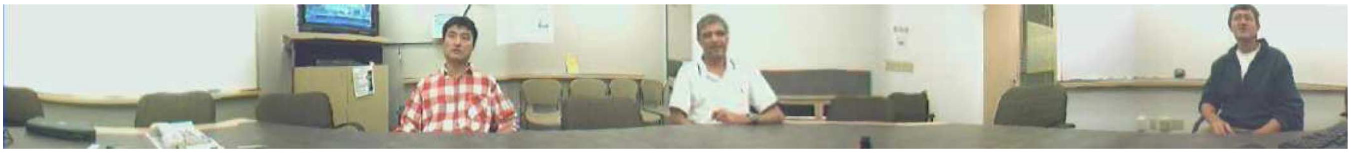


Fig. 8. After applying SVU scaling to the image in Fig. 2.

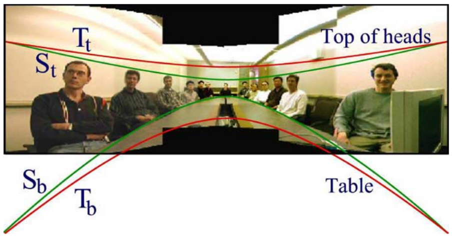


Fig. 9. Source curves and the target curves with $\alpha = 0.3$.

resolution one. At 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 $c = 1, \ldots, n$. 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 [(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 cth 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 cth 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'.

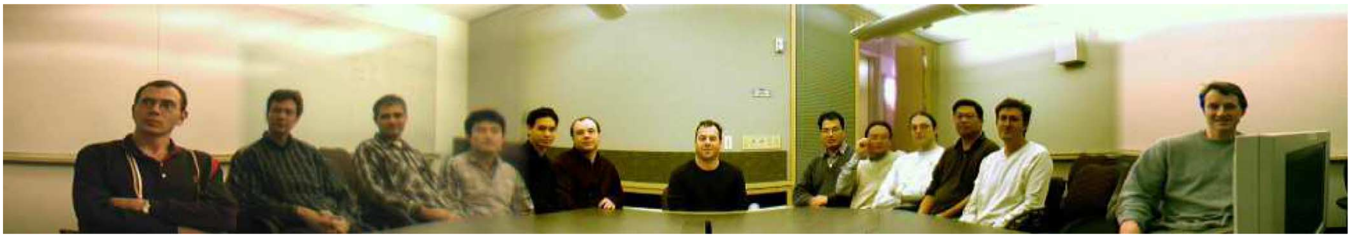


Fig. 10. SVU scaling without horizontal distortion correction.

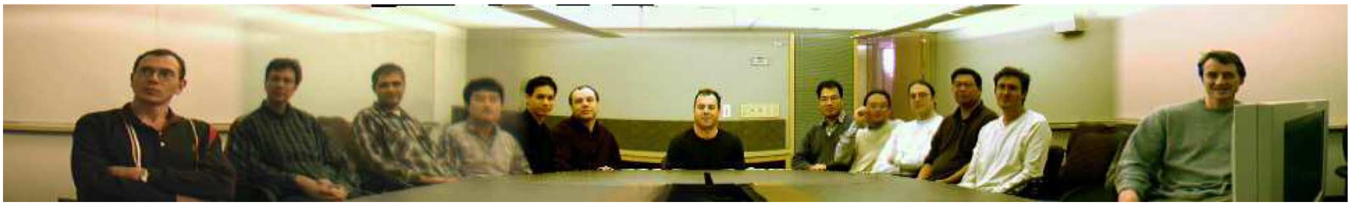


Fig. 11. SVU scaling with horizontal distortion correction.

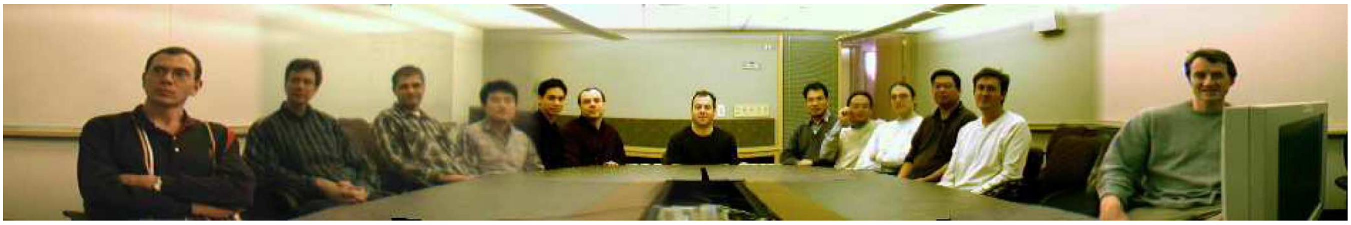


Fig. 12. SVU scaling with $\alpha = 0.2$.

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. Fig. 7 shows the two source curves on the 360-degree panoramic image as shown in Fig. 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 Fig. 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 (1) and (2). The rest of the algorithm is carried out in the same way as in the 180-degree image case. Fig. 8 shows the result after applying SVU scaling function to the image in Fig. 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

The image in Figs. 1 and 9 shows both the source and target curves with $\alpha = 0.3$. Fig. 10 through 13 show the results of using the SVU scaling function. Fig. 10 shows the result of applying the SVU scaling function without correcting horizontal distortion. Fig. 11 shows the result after correcting for horizontal distortion with $\omega = 0.8$. By comparing Fig. 11 with Fig. 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. Fig. 12 shows the result with $\alpha = 0.2$, and Fig. 13 shows the result with $\alpha = 0.4$. We would like to point out that one of the individual cameras (the second image from left in Fig. 6) does not focus well because the lens we bought has a 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 1.7-GHZ processor. The delay is approximately 65 ms.

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].

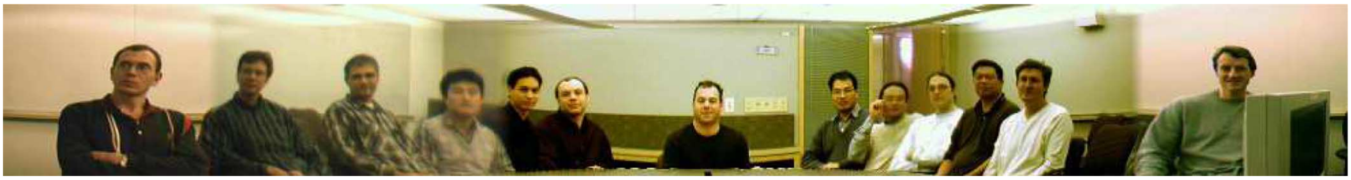


Fig. 13. SVU scaling with $\alpha = 0.4$.

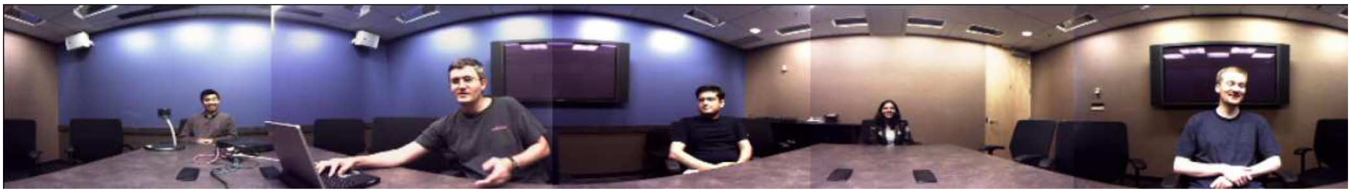


Fig. 14. Sample frame of the video captured in the first room setting. The table dimension is 12 × 5 feet.

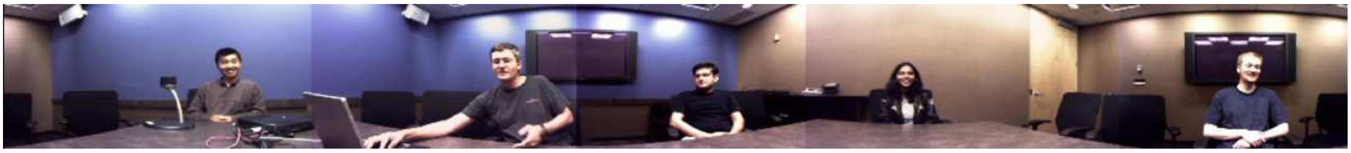


Fig. 15. Result of applying SVU scaling with $\alpha = 0.5$ to the image shown in Fig. 14.

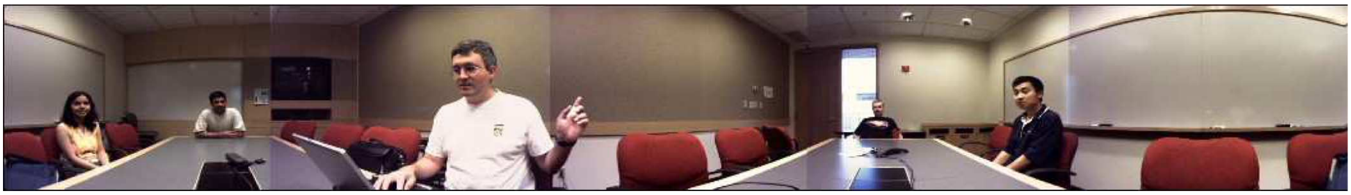


Fig. 16. Sample frame of the video captured in the second room setting. The table dimension is 16 × 5 feet.

We performed experiments in three different meeting room settings. Figs. 14, 16, and Fig. 18 each shows a sample frame from the three different room settings. For the first setting (Fig. 14), the table dimension is 12×5 feet. For the second setting (Fig. 16), the table dimension is 16×5 feet. The third setting (Fig. 16) is the same as the second setting except that a person stands up and walks around the table.

Figs. 15, 17, and Fig. 19 are the results after applying SVU scaling function with $\alpha = 0.5$ on the images in Figs. 14, 16, and Fig. 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 Figs. 14, 16, and Fig. 18. Each video is about 30 s 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 α =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. Fig. 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 Fig. 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

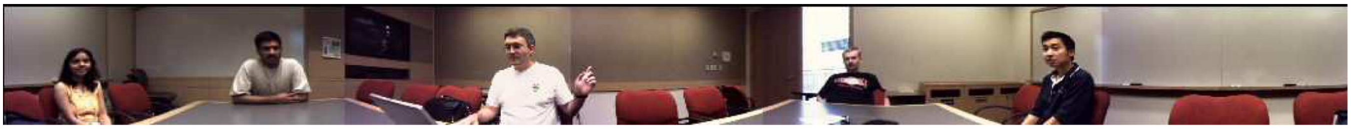


Fig. 17. Result of applying SVU scaling with $\alpha = 0.5$ to the image shown in Fig. 16.

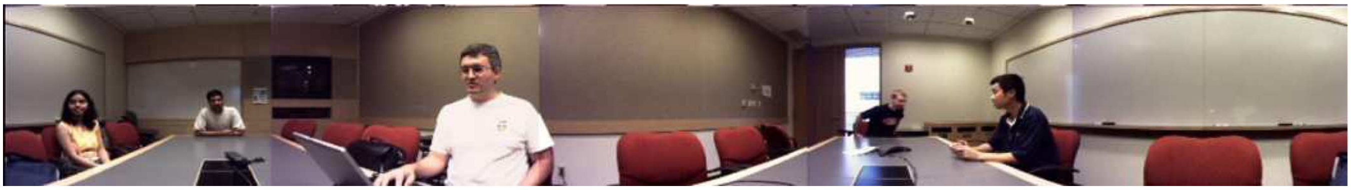


Fig. 18. Sample frame of the video captured in the third room setting where a person walks around the table.

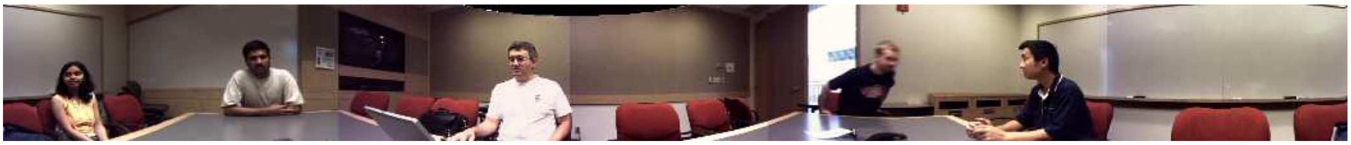


Fig. 19. Result of applying SVU scaling with $\alpha = 0.5$ to the image shown in Fig. 18.

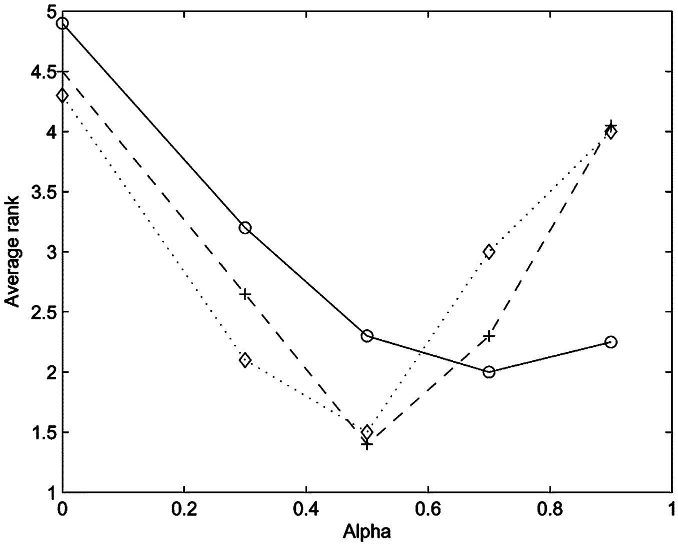


Fig. 20. Results of the user study. Solid curve: user study result for the first room setting where the meeting table size is 12×5 feet. Dashed curve: user study result for the second setting where the meeting table size is 16×5 feet. Dotted curve: user study result for the third setting where a person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

$\alpha = 0.5$ is ranked the best in both settings. The user study result suggests that $\alpha = 0.5$ is a safe choice for all table sizes.

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 *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 503–512.

[2] Y. Tamai, S. Kagami, H. Mizoguchi, K. Sakaya, K. Nagashima, and T. Takano, "Circular microphone array for meeting system," *Proc. IEEE*, vol. 2, no. 2, pp. 1100–1105, Oct. 2003.

[3] D. G.-P. H. K. Maganti and I. McCowan, "Speech enhancement and recognition in meetings with an audio-visual sensor array," *Tech. Rep. IDIAP-RR 06–24, IDIAP Res. Ins. Ecole Polytechnique Federale de Lausanne (EPFL)*, 2006.

[4] S. Nayar, "Omnidirectional video camera," in *DARPA Image Understanding Workshop*, New Orleans, LA, May 1997, pp. 235–242.

[5] S. Nayar, "Catadioptric omnidirectional camera," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, San Juan, PR, Jun. 1997.

[6] M. Aggarwal and N. Ahuja, "High dynamic range panoramic imaging," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jul. 2001, vol. 1, pp. 2–9.

[7] S. Coorg, N. Master, and S. Teller, "Acquisition of a large pose-mosaic dataset," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 872–878.

[8] S. Nayar and A. Karmarkar, "360 × 360 mosaics," in *Proc. IEEE Conf. 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, FL, 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 and F. D. L. Torre, "Automatic clustering of faces in meetings," in *Proc. IEEE Int. Conf. Image Processing*, Atlanta, GA, Oct. 2006.

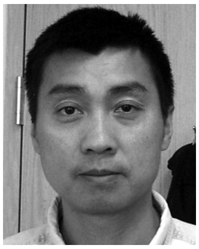
[12] Y. Chang, R. Cutler, Z. Liu, Z. Zhang, A. Acero, and M. Turk, "Automatic head-size equalization in panorama images for video conferencing," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Amsterdam, The Netherlands, Jul. 2005.

[13] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. Int. Conf. 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 *Proc. Int. Conf. Computer Vision (ICCV'95)*, 1995, pp. 605–611.

[15] S. Mann and R. W. Picard, "Virtual bellows: Constructing high quality images from video," in *Proc. 1st IEEE Int. Conf. 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 *Comput. Graph.—Annu. Conf. Series*, 1997, pp. 251–258.



Zicheng Liu (SM'05) received the B.S. degree in mathematics from Huazhong Normal University, Wuhan, China, the M.S. degree in operational research from the Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, and the Ph.D. degree in computer science from Princeton University, Princeton, NJ.

He is a Researcher at Microsoft Research, Redmond, WA. Before joining Microsoft, he was a Member of Technical Staff at Silicon Graphics, focusing on trimmed NURBS tessellation for CAD model visualization. His research interests include linked figure animation, face modeling and animation, face relighting, image segmentation, and multimedia signal processing.

Dr. Liu is an Associate Editor of *Machine Vision and Applications*. He was a Co-Chair of the 2003 IEEE International Workshop on Multimedia Technologies in E-Learning and Collaboration, Nice, France, a Program Co-Chair of the 2006 International Workshop on Multimedia Signal Processing (MMSP), Victoria, Canada, and an Electronic Media Co-Chair of the 2007 International Conference and Multimedia and Expo, Beijing, China.

Michael Cohen (SM'03) received the B.S. degree in art from Beloit College, Beloit, WI, the B.S. degree in civil engineering from Rutgers University, New Brunswick, NJ, the M.S. degree in computer graphics from Cornell University, Ithaca, NY, and the Ph.D. degree from the University of Utah, Salt Lake City, in 1992.

He previously served on the Architecture Faculty at Cornell University, was an Adjunct Faculty Member at the University of Utah, and was a member of the Faculty of Computer Science at Princeton University, Princeton, NJ, He joined Microsoft Research, Redmond, WA, in 1994, where he is currently a Principal Researcher. His early work at Cornell and Princeton on the radiosity method for realistic image synthesis is discussed in his book (co-authored by J. R. Wallace) *Radiosity and Image Synthesis* (New York: Kauffman, 1993). His work at the University of Utah focused on space–time control for linked figure animation. At Microsoft, he has worked on a number of projects ranging from image-based rendering to animation, to camera control, to more artistic nonphotorealistic rendering. One project focuses on the problem of image-based rendering: capturing the complete flow of light from an object for later rendering from arbitrary vantage points. This work, dubbed The Lumigraph, is analogous to creating a digital hologram. He has also continued his work on linked figure animation, focusing on means to allow simulated creatures to portray their emotional state. Recent work has focused on computational photography applications, ranging from creating new methods for low-bandwidth teleconferencing, segmentation and matting of images and video, technologies for combining a set of "image stacks" as a photomontage, to the creation of very high-resolution panoramas.

Dr. Cohen received the 1998 SIGGRAPH Computer Graphics Achievement Award for his contributions to the Radiosity method for image synthesis. He served as Paper's Chair for SIGGRAPH '98.

Deepti Bhatnagar is currently pursuing the B.S. degree at the Indian Institute of Technology, Delhi, India.

She was a summer intern at Microsoft Research, Redmond, WA, during 2005. Her areas of interest are computer vision, machine learning, and multimedia.

Ross Cutler received the B.S. degree in mathematics, computer science, and physics and the M.S. and Ph.D. degrees in computer science, all from the University of Maryland at College Park.

He is currently a Principal Architect at Microsoft Research, Redmond, WA. His research interests include computer vision, video indexing, multimedia databases, multimedia authoring, multiview imaging, gait recognition, face recognition, and real-time systems.

Zhengyou Zhang (SM'97–F'05) received the B.S. degree in electronic engineering from the University of Zhejiang, Zhejiang, China, in 1985, the M.S. degree in computer science (specializing in speech recognition and artificial intelligence) from the University of Nancy, Nancy, France, in 1987, the Ph.D. degree in computer science (specializing in computer vision) from the University of Paris XI, Paris, France, in 1990, and the Dr. Sci. (Habil. diriger des recherches) diploma from the University of Paris XI in 1994.

He is a Principal Researcher with Microsoft Research, Redmond, WA, which he joined in 1998. He has been with INRIA (the French National Institute for Research in Computer Science and Control) for 11 years and was a Senior Research Scientist from 1991 to 1998. During 1996–1997, he spent a one-year sabbatical as an Invited Researcher at the Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan. He has published over 150 papers in refereed international journals and conferences, and has co-authored the following books: *3D Dynamic Scene Analysis: A Stereo Based Approach* (Berlin/Heidelberg: Springer, 1992); *Epipolar Geometry in Stereo, Motion and Object Recognition* (Norwell, MA: Kluwer, 1996); *Computer Vision* (textbook in Chinese, Chinese Academy of Sciences, 1998).

Dr. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, an Associate Editor of the *International Journal of Computer Vision* (IJCV), an Associate Editor of the *International Journal of Pattern Recognition and Artificial Intelligence* (IJPRAI), and an Associate Editor of *Machine Vision and Applications* (MVA). He served on the Editorial Board of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI) from 2000 to 2004, and of two other journals. He has been on the program committees for numerous international conferences, and was an Area Chair and a Demo Chair of the International Conference on Computer Vision (ICCV2003), October 2003, Nice, France, a Program Co-Chair of the Asian Conference on Computer Vision (ACCV2004), January 2004, Jeju Island, Korea, and a Demo Chair of the International Conference on Computer Vision (ICCV2005), October 2005, Beijing, China. He delivered a keynote speech at the International Workshop of Multimedia Signal Processing (MMSP), Shanghai, November 2005, and was a Program Co-Chair of the International Workshop on Multimedia Signal Processing (MMSP), Victoria, Canada, October 2006.