Gaze Awareness for Videoconferencing: A Software Approach

Jim Gemmell and Kentaro Toyama
Microsoft

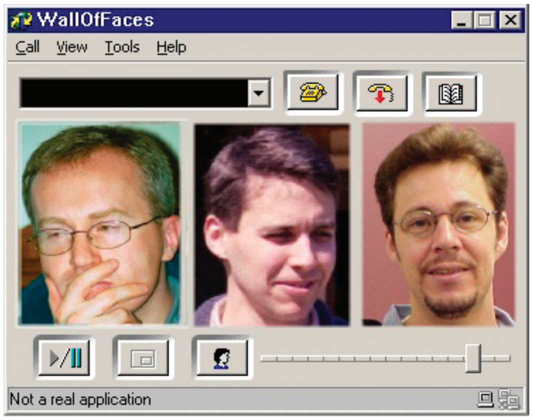
C. Lawrence Zitnick and Thomas Kang
Carnegie Mellon University

Steven Seitz
University of Washington

Previous attempts at bringing gaze awareness to desktop videoconferencing have relied on hardware solutions. Here, the authors describe their software approach, which tracks participants' head and eye movements using vision techniques, then uses this information to graphically place the head and eyes in a 3D environment.

The impact of gaze is striking. We all know the experience of "feeling watched," and it's hard to resist the urge to look at someone who's staring at you. As the sidebar, "Human Gaze: A Closer Look," explains, gaze awareness—and eye contact in particular—is extremely important in face-to-face communication. In addition to practical uses, such as using gaze to direct turn-taking in conversation, gaze awareness has more abstract social value: People who use frequent eye contact are perceived as more attentive, friendly, cooperative, confident, mature, and sincere than those who avoid it.

Despite the obvious importance of directed gaze, most videoconferencing systems make it impossible for participants to make eye contact or even to determine where or at what the other participants are looking. This loss of gaze awareness has a profound impact on communication, and may in fact be a contributing factor in the failure of videoconferencing to meet with its long-anticipated success (for more on this, see the sidebar, "Videoconferencing Research" on page 28).



Not a real application

Figure 1. The typical videoconferencing interface doesn't provide gaze awareness or give participants a sense of spatial relationships.

The lack of gaze awareness in typical videoconferencing systems stems from the fact that when participants look at each other, they stare into their displays rather than into the camera, which is typically mounted above, below, or beside the display. Unless people look directly into the camera, you will never perceive them as making eye contact with you, no matter where you're situated in relation to the display. Conversely, if they're looking into the camera, they will always appear to be looking at you, even when you move around. A famous example of this is Leonardo da Vinci's painting of the Mona Lisa, whose eyes appear to follow viewers around the room.

As Figure 1 shows, videoconferencing typically gives each participant an individual, arbitrarily placed window, and participants never appear to look at each other. In addition to inhibiting eye contact between two parties, with multiparty desktop systems participants have no sense of spatial relationship and cannot tell who's looking at whom. Existing systems restore gaze awareness only by using special and typically expensive hardware.

We developed a software-based approach to videoconferencing gaze awareness. We aim to develop an inexpensive videoconferencing system that supports two to four participants. To date, hardware has been a serious obstacle to videoconferencing. Thus, it was important that our system work with cheap, commonly available hardware. The system uses standard PC hardware, equipped with audio I/O and video capture. Virtually all PCs now ship with sufficient audio support, and many are beginning to ship with video capture. For those

1070-986X/00/$10.00 © 2000 IEEE

Human Gaze: A Closer Look

In basic human interactions, people tend not to gaze fixedly into each other's eyes. Typically, people gaze at points around the area of another's eyes and mouth, shifting points about every one-third second.[1] Mutual gaze is typically held for no more than one second and doesn't require actual eye contact—it can be directed anywhere in the facial region. In fact, most people have difficulty detecting actual eye contact, whereas they're fairly accurate at detecting face-directed gaze.[1,2] As distance between people grows, their accuracy in detecting gaze decreases. Accuracy also declines when people aren't directly facing each other.[1,3,4] Researchers have surmised that pupil position is most likely the primary determinant of gaze direction.[5]

Human perception of head pose seems heavily influenced by the region around the eyes and nose—perhaps because people tend to direct their gaze there. Figure A illustrates this idea. Although the subject's head is oriented toward the viewer, we superimposed a cutout of the eyes and nose that are turned to the right. At first glance, the entire head seems oriented to the right. Researchers also conducted an experiment that changed room lighting[6] and found that visible parts of a face's left and right profile lines might give people important cues in perceiving head pose.

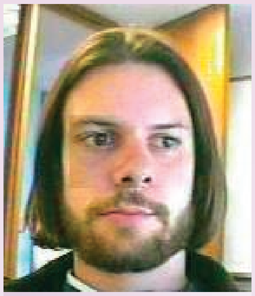


Figure A. Here, we superimposed the eyes and nose onto the face—at first glance, the head appears to be turned to the right.

The social importance of eye contact has long been established.[1] For example, people definitely notice the difference between receiving direct gaze 15 percent of the time versus 85 percent of the time. Experiments have found that people using eye contact received more job offers after interviewing, more help after asking for it, and are generally considered more persuasive. Teachers using eye contact have more productive students who learn faster than other students. People who use eye contact also appear friendly, self-confident, natural, mature, and sincere, while those who don't seem cold, pessimistic, cautious, defensive, immature, evasive, submissive, indifferent, and sensitive. Also, individuals look at each other more when they cooperate than they do when they compete.[1]

Speakers glance at listeners to elicit response and, more importantly, to obtain information about listeners through expressions, head nods, and other signals. Speakers also widen their eyes to emphasize points. Gaze direction can also be important for silent participants, such as when they direct a smile or wink at another participant. Listeners typically focus their gaze on the speaker 70 to 75 percent of the time in seven- to eight-second glances. Among the things listeners look for are nonverbal communication through body language, expression, and so on, and they also read the speakers lips on occasion (a clear view of the lips can make up for several decibel of noise).[1]

Finally, when individuals meet in groups of three, gaze is typically divided between the other two parties and mutual gaze occurs only about 5 percent of the time. In this situation, gaze coordinates turn-taking in conversation, but isn't always the only or most important cue.[7]

References

1. M. Argyle, *Bodily Communication,* International Universities Press, Madison, Conn., 1988.
2. K. Morii et al., "A Technique of Eye Animation Generated by CG, and Evaluation of Eye-Contact Using Eye Animation," *Proc. Int'l Society for Optical Engineering (SPIE),* Vol. 1818, Pt. 3, SPIE Press, Bellingham, Wash., 1992, pp.1350-1357.
3. B. Rime and L. McCusker, "Visual Behaviour in Social-Interaction—Validity Of Eye-Contact Assessments Under Different Conditions of Observation," *British J. of Psychology*, Vol. 67, No. 5, 1976, pp. 507-514.
4. A.M. Noll, "Effects of Visible Eye and Head Turn on Perception of Being Looked At," *American J. of Psychology*, Vol. 89, No. 4, 1976, pp. 631-644.
5. S.M. Anstis, J.W. Mayhew, and T. Morley, "The Perception of Where a Face or Television 'Portrait' is Looking," *American J. of Psychology*, Vol. 82, No. 4, 1969, pp. 474-489.
6. N.F. Troje and U. Siebeck, "Illumination-Induced Apparent Shift in Orientation of Human Heads," *Perception*, Vol. 27, No. 6, 1998, pp. 671-680.
7. D.G. Novick, D.G. Hansen, and K. Ward, "Coordinating Turn-Taking with Gaze," *Proc. Int'l Conf. on Spoken Language Processing (ICSLP'96)*, Alfred I. duPont Institute, Wilmington, Del., 1996, pp. 188-191.

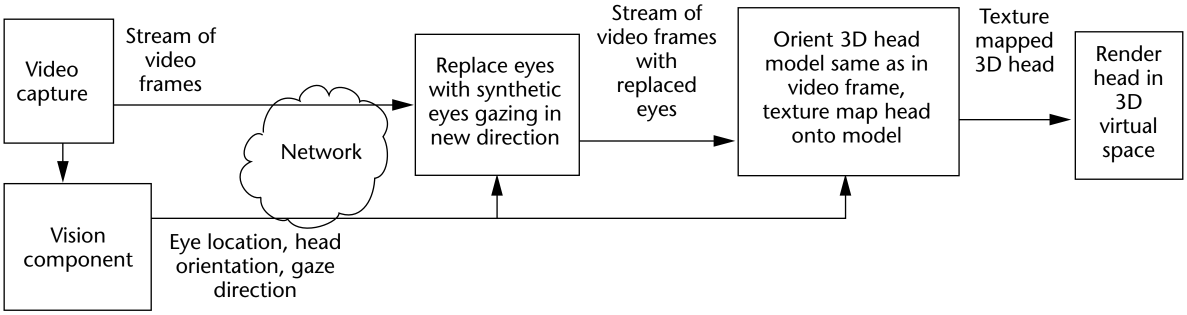

Video capture
Stream of video frames
Network
Replace eyes with synthetic eyes gazing in new direction
Stream of video frames with replaced eyes
Orient 3D head model same as in video frame, texture map head onto model
Texture mapped 3D head
Render head in 3D virtual space
Vision component
Eye location, head orientation, gaze direction

Figure 2. An overview of our video subsystem for supporting gaze awareness.

that don't, you can add a universal serial bus camera or a camera and capture card, both of which are inexpensive. The full system supports 3D surround-sound audio to better position participants' voices in the virtual space. We might also add other features, such as a whiteboard and application sharing. Here, we focus our discussion on our video subsystem for supporting gaze awareness.

Figure 2 shows the architecture of our video subsystem. As in traditional videoconferencing, the sys-

Videoconferencing Research

Videoconferencing began with Ives' 1927 research at Bell labs.[1] Since then, videoconferencing has been repeatedly hailed as on the brink of ubiquity: with the unveiling of the PicturePhone at the 1964 World's Fair, with the introduction of Integrated Services Digital Network videoconferencing in the 1980s, and the arrival of cheap desktop videoconferencing in the 1990s. However, the technology has never caught on as well as expected.

Some studies of group problem solving and task accomplishment found no advantage in videoconferencing over audio-only communication. Chapanis and his colleagues compared problem solving using face-to-face communication to that using voice only, handwriting, and typing.[2] They found that for problem solving, voice is only a little slower than face-to-face, and everything else takes at least twice as long. The fact that voice is nearly as fast as face-to-face seems to imply that video is unnecessary. In a similar study, Gale compared data sharing, data sharing with audio, and data sharing with audio and video and found no difference in task completion time or quality.[3] Sellen also found no significant contribution from video.[4] Acker and Levitt found that people were happy with gaze-aware videoconferencing as a medium, but that it did not improve the final outcome of the given task.[5]

Based on such studies and their own negative personal experiences, some researchers have concluded that videoconferencing isn't worthwhile. However, the negative results are in part due to technical problems—many systems have suffered from audio latencies and difficult call setup. Furthermore, it's not clear that solving contrived problems is the true test of videoconferencing. If video contributes to enhanced communications, it could prove valuable in other contexts, such as negotiations, sales, and relationship building. In fact, within these very studies, researchers observed that the "rate of social presence" is increased by video[3] and that people took advantage of gaze awareness in a system that supported it.[4]

Several studies suggest that replicating the conversation process requires spatial audio and video.[6,7] AT&T redesigned PicturePhone in the late 1960s, with the goal of reducing "eye-contact angle," which they believed is perceptible after about five degrees.[1] A study of a point-to-point system with two or four people at each end found that gaze correction improved participants' perception of nonverbal signals.[8]

Among the hardware techniques that support gaze-aware videoconferencing systems are half-silvered mirrors and pinhole cameras in displays.[5,9] The Virtual Space and Hydra systems support gaze awareness by deploying a small display and camera for each party. If you place the display far enough away from users, they're unlikely to notice the angle between the camera and the display images.[4,6]

Although the Teleport system doesn't support gaze awareness,[10] its creators note this problem and recommend warping images onto 3D models, as we have done in our system.

tem captures a stream of video frames and transmits it across the network. A vision component analyzes the video frames and determines eye contour, head orientation, and gaze direction. The system then transmits this information across the network with the video frames. At the receiving end, the system uses the video frames and vision information to render the head, with the desired gaze, in a virtual 3D space. In the following sections, we elaborate on the rendering and vision components, followed by an outline of our future plans.

Rendering

On the receiving end, the system uses the vision information to extract the head from the video frame, correct the gaze, and place it in the virtual 3D space. We achieve this in two steps. First, the system replaces the eyes with synthetic eyes to aim the gaze. The synthetic eyes simulate eyes swiveling in their sockets. Second, the system adjusts the head pose. To achieve the desired gaze, the eye replacement must account for the forthcoming head pose adjustment.

Alternatively, we could use an entirely synthetic head, or avatar. We have a spectrum of possibility here, from using unmodified video to using a fully synthetic avatar. Our approach lies between these extremes. The benefit of our approach is that we transmit facial expressions and eye blinks as they appear, while modifying only those aspects of video that affect gaze perception. To achieve a similar effect with an avatar, we would need a very detailed head model and detailed tracking of additional facial points. As we discuss below, even our modest tracking requirements still require more research—tracking many facial points is not cur-

Avatars (fully synthetic characters) have also been used for teleconferencing. One system[11] tracks the viewer's facial features using tape marks attached to the viewer's face, and detects real-time movements in the head, body, hands, and fingers using magnetic sensors and data gloves. Colburn and his colleagues are investigating the use of eye gaze in avatars.[12] They found that viewers respond to avatars that have natural eye-gaze patterns by changing their own gaze patterns, which helps draw attention to different avatars.

References

1. R. Stokes, "Human Factors and Appearance Design Considerations of the Mod II PicturePhone Station Set," *IEEE Trans. Communication Technology*, Vol. COM-17, No. 2, April 1969, pp. 318-323.
2. A. Chapanis et al., "Studies in Interactive Communication: I. The Effects of Four Communication Modes on the Behaviour of Teams During Cooperative Problem-Solving," *Human Factors*, Vol. 14, No. 6, 1972, pp. 487-509.
3. S. Gale, "Adding Audio and Video to an Office Environment," *Studies in Comp. Supported Cooperative Work*, J.M. Bowers and S.D. Benford, eds., Elsevier Science Publishers, New York, 1991, pp. 49-62.
4. A. Sellen, "Remote Conversations: The Effects of Mediating Talk with Technology," *J. Human-Computer Interaction*, Vol. 10, No. 4, 1995, pp. 401-444.
5. S.R. Acker and S.R. Levitt, "Designing Videoconferencing Facilities for Improved Eye Contact," *J. Broadcasting and Electronic Media*, Vol. 31, No. 2, Spring 1987, pp. 181-191.
6. G.M. Hunter, "Teleconference in Virtual Space," *Information 80: Proc. Eighth World Comp. Congress*, S. Lavington, ed., North Holland, Amsterdam, 1980, pp. 1045-1048.
7. B. O'Connaill and S. Whittaker, "Characterizing, Predicting and Measuring Video-Mediated Communication: A Conversational Approach," *Video-Mediated Communication*, K. Finn, A. Sellen, and S. Wilbur, eds., Lawrence Erlbaum Associates, Hillsdale, N.J., 1997.
8. A. Suwita et al., "Overcoming Human Factors Deficiencies of Videocommunications Systems by Means of Advanced Image Technologies," *Displays*, Vol. 17, No.2, April 1997, pp. 75-88.
9. D.A.D. Rose and P.M. Clarke, "A Review of Eye-to-Eye Videoconferencing Techniques," *BT Technology J.*, Vol. 13, No.4, Oct. 1995, pp. 127-31.
10. S.J. Gibbs, C. Arapis, and C.J. Breiteneder, "Teleport—Toward Immersive Copresence," *Multimedia Systems*, Vol. 7, No. 3, 1999, pp. 214-221.
11. J. Ohya et al., "Real-Time Reproduction of 3D Human Images in Virtual Space Teleconferencing," *Proc. Virtual Reality Annual Int'l Symp.,* IEEE Comp. Society Press, Los Alamitos, Calif., 1993, pp. 408-414.
12. A. Colburn, M. Cohen, and S. Drucker, "The Role of Eye Gaze in Avatar Mediated Conversational Interfaces," Tech Report MSR-TR-2000-81, Microsoft Research, Redmond, Wash., July 2000.

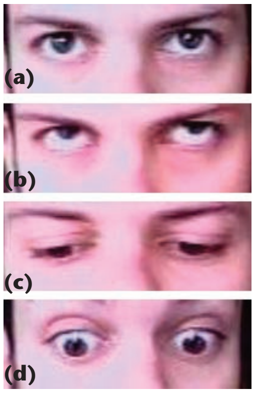


Figure 3. As the eyes move up and down, so do the eyelids. If the system ignores this, facial expressions can change. (a) The subject looks up; (b) when the system does not adjust the eyelids accordingly, the subject appears to be glowering or disgusted. (c) The subject looks down; (d) without eyelids adjustment, the subject appears surprised.

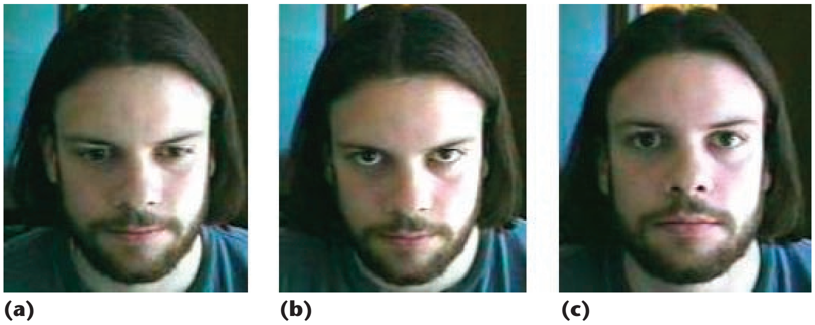


Figure 4. (a) The subject's head and eyes are directed away from viewer, showing a lack of gaze awareness and general disinterest. (b) When the eyes are directed at the viewer and the head is directed away, it creates eye contact but gives the face a distrustful or disapproving expression. (c) When the subject's head and eyes are directed at the viewer, it creates eye contact and an attentive facial expression.

rently feasible. A drawback of our approach is that it can produce some distortion of the face, which wouldn't occur with detailed head models.

It's also possible to achieve arbitrary gaze direction solely by manipulating the head pose. However, for this work, the head must swivel every time the gaze changes. Depending on the virtual space's geometry in relation to the geometry of the viewer and the display, this can create a lot of virtual head movement. In particular, viewers who move their eyes back and forth between two images might appear to be shaking their heads and thus indicating "no" in their virtual representations.

Likewise, gaze can be corrected solely by adjusting the eyes. However, simply repositioning the pupils can change facial expression. As Figures 3a and 3c show, when a person looks up or down, the top eyelid typically follows the top of the pupil. If a system fails to synthesize this change in the eye area, the eyelid might appear too low, giving the face an expression of disgust (Figure 3b), or too high, creating a surprised expression (Figure 3d). Horizontal changes in pupil position have little noticeable effect on facial expression.

Head orientation itself also conveys a message: a lowered head might indicate distrust or disapproval, while a raised head might convey superiority. Facial expression is further affected by how far the eyes are opened and the amount of white showing above and below the pupil.[1] Figure 4 shows this to some extent, though to get the full impact you have to see the actual motion of raising and lowering. A correction of the vertical head pose angle between the camera and the images onscreen is required to convey an accurate message to the viewer.

Eye manipulation

Our system manipulates the eyes in two steps. First, the vision component segments the eyes. As we discuss in detail below, the vision component indicates the video frame region that contains the visible portion of the eyeballs. In the second step, we use computer graphics to render new (synthetic) eyes focused on the desired point in space.

Many well-known computer graphics techniques exist for eye synthesis, including some that are sophisticated and very realistic. However, we use a relatively simple technique. We assume that the average color of the sclera ("white" of the eye), iris, and pupil are known. If we know the size of the eyeball, we can estimate the relative size of the iris. We fix the pupil's radius to be a fraction of the iris's radius. We don't currently model pupil dilation and contraction. To simplify rendering, we also model eyes without curvature. In practice, this proves a reasonable choice because eye curvature is only noticeable when the subject's head is significantly turned away from the viewer (more than 30 degrees from our observations).

We begin with a background that matches the sclera's average color value. The system then draws two circles representing the iris and pupil. Next, the system draws another circle the color of the pupil around the edge of the iris to represent the limbus (the iris's dark outer edge). The system adds random noise to the iris and the sclera to simulate texture. In a high-resolution system, we might switch to a more elaborate eye model with improved shading, highlights, and spectral reflections.

The system draws the eyeball on the face in two steps. First, the system draws the eyeballs on a temporary image (see Figure 5a). Next, the system uses the eye segmentation data to decide, for each pixel, whether to use the pixel information from the original face image or from the eyeball image. The simplest method for combining the face image and eyeball image uses color keying, which resembles blue screening. As Figure 5b shows, we color each segmented eyeball pixel blue (or whichever color is the color key color). We can then blit the eyeball image onto the face image. For a more refined or realistic look, we can use alpha values to blend the eyeball's edges with the face image (see Figure 6).

Controlling eye gaze means controlling where the eyes are looking in 3D space. The 3D point that the eyes are looking at is called the gaze point. Our goal is to determine pupil positions that give the appearance that the eyes are focused on the desired gaze point. The eyes should converge or diverge as the gaze point moves closer or further away from the face. To compute the pupil positions, we must determine (1) the 3D location of the eyeball centers relative to the head model, (2) the eyeball radius, and, if we have a 3D head model, (3) its orientation and position in 3D space.

Because our eyeball model is flat, we need only compute the plane on which to render the eye and the pupil's center. We select the plane to lie in the eye sockets, oriented perpendicular to the gaze direction. We find the pupil's center by intersecting the ray from the gaze point to the eyeball center with the eye plane. Approximating the flat eyeball becomes increasingly inaccurate as the head rotates away from the viewer. However, this inaccuracy is mitigated by two factors. First, extreme combinations of head orientation and eye gaze (such as a head facing left and eyes gazing sharply to the right) are rare. They're also difficult to modify for other reasons—including that tracking the eyes in such situations presents significant challenges for computer vision. Given this, our project restricts eye-gaze modification to instances when the head is oriented frontally (within about 30 degrees). Second, studies have shown that humans are poor judges of eye gaze when a subject isn't directly facing us,[2-4] and thus the errors in approximating extreme poses are unlikely to bother viewers.

Altering head pose

Our first attempt to rotate the head involved warping the face image using correspondence maps. However, we found many difficulties with this method, including unacceptable distortions.[5] Our present approach changes the head orientation using texture mapping. First, the system creates a 3D model in the shape of the subject's head.

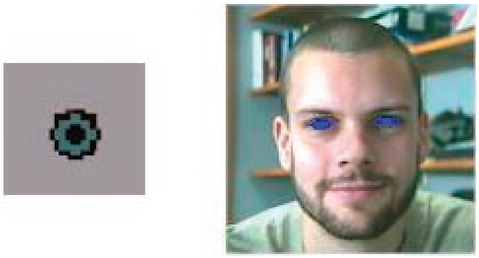


Figure 5. (left) A synthetic eyeball on a temporary image and (right) a video frame with the segmented eyeball area filled with blue color.

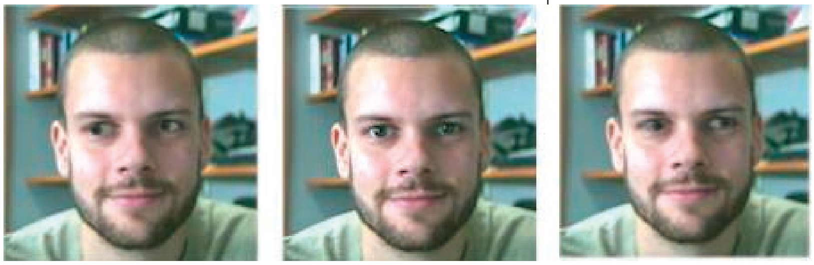


Figure 6. Original image (center) and two synthesized images with redirected eyeballs looking left (left) and looking right (right).

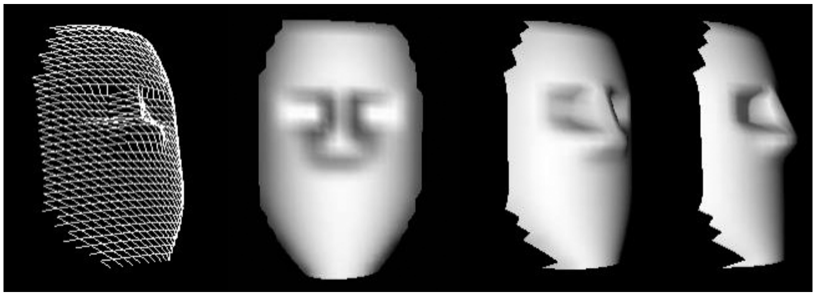


Figure 7. Simple head model used for texture mapping.

Next, as Figure 7 shows, the system executes texture mapping, projecting the face image from the video frame onto the model. It can then rotate the model to any desired orientation (see Figure 8, next page).

To texture map a model, we must know three values:

1. The position of an anchor point on the 3D model and its location in the face image. We use the center between the nostrils as the anchor point. This point is easy to track because it doesn't deform significantly when the face rotates or the expression changes.

2. The orientation of the head in the video frame.

3. How to scale the head model to correspond to pixel space.

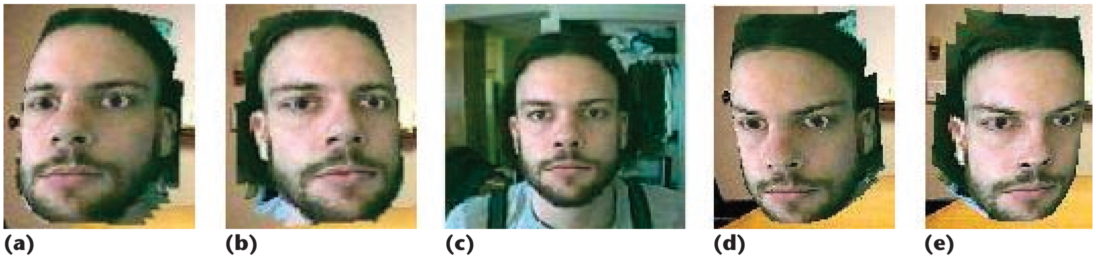


Figure 8. After texture mapping, the system can rotate the model in any direction. (a) Face oriented up left, (b) face oriented up right, (c) original image, (d) face oriented down left, and (e) face oriented down right.

For each vertex of the head model, we must compute its 2D texture coordinates. The texture coordinates define the location in the face image corresponding to the vertex. If we have the three values described above, we can compute the texture coordinates for each vertex with the following steps. Here, we assume that we're rotating the model using the nose as an anchor point. However, any point may be used.

- For each vertex in the head model, subtract the value of the anchor point.
- Rotate the head model to the same orientation as the head in the face image.
- Scale the X and Y coordinates to correspond to pixel values. This is equivalent to doing an orthographic projection onto the image. An orthographic projection assumes all lines of sight are parallel (unlike the pinhole projection, in which the lines of sight intersect at a point).
- Add the 2D pixel location of the nose in the face image.

When creating a 3D head model, certain details are more important than others. Based on our experience, we identified two key features for judging head orientation. First, we must model the eyes correctly. Although the eyeballs themselves are flat, the eye socket must recede into the face. This is important for creating a realistic look when the head rotates up and down. Second, we must model the nose so that it protrudes from the face.

We model facial parts that have less affect on head orientation—such as the mouth, forehead, and cheeks—as flat or slightly rounded surfaces. To account for differences in facial shape, we separately fit the model to each subject. Because the eyes and nose are the most important features, we scaled the model to fit the face based on their geometric relationship. We also fixed the amount that eyes recede and the nose protrudes. Once again, we assume that the head will not be rotated more than 30 degrees, so the results should be realistic for a reasonable range of facial orientations.

When a person talks or changes expression, the head's 3D shape can change. The most obvious example is that when people talk, the chin moves up and down. Given this, we assume that the shape of a subject's head model will change. To deal with this, we extend the wireframe chin below its normal position. This way, when the mouth opens, the chin texture won't slip down to the neck area. When the mouth closes, the neck will look like it's attached to the bottom of the chin, but this should only be noticeable when the head is rotated significantly away from the viewer.

As we described above, people typically judge head orientation using the eyes and nose. This might be because both are relatively static—eye sockets don't change shape, and noses rarely deform. Our model takes advantage of this, letting us use a static head model to achieve reasonable realism.

Facial features are increasingly deformed as the difference in rotation between the rendered head and the head in the video is increased. Assuming that we found correct texture coordinates, such facial deformations result from an incorrect face model. Many deformations go unnoticed, such as the side of the head being too narrow. Other deformations can actually cause changes in facial expression. If a head model has an incorrect mouth curvature, for example, the subject's mouth might look either happy or sad when the head model rotates up and down. Similar changes in expression can result when eyebrows are inaccurately curved. Because every person has a distinct mouth curvature, we might solve this problem by using "structure from motion" algorithms from vision, which compute the 3D structure of a face using a video sequence.[6] Identifying which parameters are important to ensuring consistent expressions is in itself a difficult task.

Unintended changes in expression and distortions also occur when the vision component gives an inaccurate head location and orientation, resulting in a misalignment of pixels on the head model. This problem is most noticeable with errors in the vertical orientation/position of the head.

Computer vision

Our system's vision component must track the head pose (the head's location and orientation), segment the eyes, and determine gaze direction. Computer vision in general is very difficult, and these tasks are no exception. However, several factors make the problem more tractable in our system:

- we're looking only for a head, not arbitrary objects;
- we deal only with head poses that permit the subject to gaze directly at the screen (if the head is turned further away, it's not looking at anyone on screen, and we simply display the image without alteration);
- the system only needs to detect gaze that is directed at the screen;
- similarly, because humans perceive any gaze around their face as mutual gaze, pin-point gaze location is not required; the system only needs to detect gaze in the facial region; and
- humans make errors—such as misjudging gaze direction as the head angle increases and misjudging head pose based on the visible head outline—and thus some system errors are likely to be no different than what a human would perceive.

We tried two different approaches to tracking head pose. The first approach assumes that the camera is mounted below the display and tracks the user's nostrils.[5] We compute head orientation based on nostril deformation. We then track nose position using standard template-matching techniques. A nose template is created from the subject's initial nose position. In each successive video frame, we search around the nose's last location to find the best template match. When the head changes orientation, the nostril shape deforms. To predict the nostril deformation, we create a set of nostril templates for a range of head orientations using an affine transformation of the tracking template. The nose tracker matches these nostril-orientation templates to the current face image to find head orientation. This approach has yielded good results, but can have trouble with bushy moustaches or if subjects wrinkle their noses.

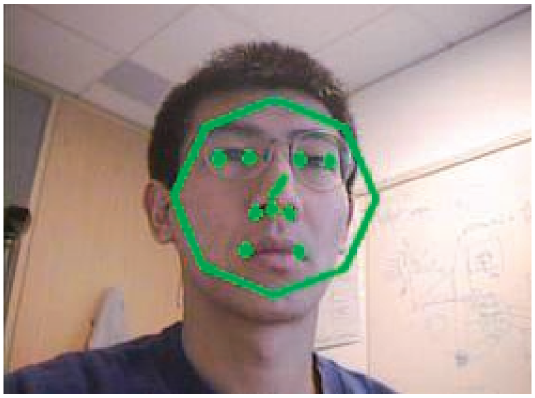


Figure 9. A single frame of face-pose tracking shows the nine tracked points on the face, an octagon indicating the face's plane, and a line indicating a normal to the face.

Our second approach determines head pose by tracking nine small image features on the face (see Figure 9). This approach assumes that we know each feature's approximate 3D position relative to the face (again, we get this information using motion techniques to determine structure). Each image feature consists of a small rectangular template whose default size occupies 8 × 8 pixels in the image (we can warp the template according to the expected size and orientation of the face in the image). For each new frame, we search for the minimum sum of pixel-wise absolute differences (SAD) between the template and each subrectangle within a restricted region of the live image. We start the matching at a coarse resolution and proceed to finer resolutions to reduce computation. Once we determine the positions for all nine image features, we perform a gradient descent in the six-degree-of-freedom pose space (x, y, z, pitch, roll, and yaw) to estimate the final pose. The goal is to minimize the sum of distances between the tracked points and the projected position of those points given a particular pose. Using the last known pose as a starting point, Levenberg-Marquardt optimization[7] achieves this goal within a handful of iterations. We use the projected positions of the feature points as the centers of the search regions for tracking in the next frame.

Pose tracking can fail if a large portion of the subject's face is occluded or if the subject moves too quickly. To maintain robustness in these circumstances, we recover tracking quickly using a fast attentional mechanism.[8] The system detects

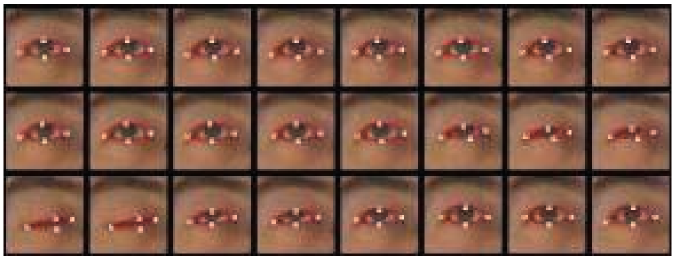


Figure 10. This data set tests the warp tracker's ability to track changes in shape. The upper left image is the canonical image, for which we manually initialized the contour. The warp tracker automatically generated the remaining image contours.

feature-tracking failure when a large residual error occurs in the SAD computations. It then resorts to skin-color "blob tracking" and head contour tracking to localize the face and reinitialize feature tracking. When the feature trackers find their respective targets, pose tracking continues.

We also tried two approaches to eye segmentation. The first builds on a deformable "active contour,"[9] which tracks the contour formed by the upper and lower eyelids. Once eye position is determined (based on head-pose information), the system processes rectangular regions centered on each eye to enhance the eye contours using histogram equalization, smoothing, and ridge detection. The system then estimates the best position for the eyelid contour using coordinate descent in the space of parameters corresponding to translation in x, translation in y, width, in-plane rotation, scale, and eye openness.

Our second approach to eye segmentation is the "warp tracker" system,[10] which tracks features through a sequence of images. Our software initializes the warp tracker using an initial (perhaps canonical) image of the eyes and a contour around the visible eye area. This image and contour define the source; eye segmentation then amounts to tracking the source contour. For any subsequent target image, we compute the correspondence map between the source and target images and apply it to the source contour to yield the target contour. To find the correspondence map, we use an automated multiresolution lattice deformation technique. In our experience, the warp tracker performs fairly well on its own, and its hierarchical nature naturally lends itself to integration with other tracking algorithms for increased accuracy and robustness. Figure 10 shows some of our warp tracker results. While the early results are promising, it's still too slow (2 frames per second on a Pentium 333) and sometimes produces irregularly shaped contours.

Conclusion and future work

The results of our work to date appear promising. Given that we can extract accurate vision data from each video frame regarding head pose, eye segmentation, and gaze direction, we can arbitrarily position and pose the head in a virtual 3D space, and synthesize the eyes with the appropriate gaze direction. The resulting videoconferencing system supports a sense of space and gaze awareness.

Currently, our chief difficulty is with the vision component. Our current methods are still too slow and inaccurate. Work on the warp tracker continues at Carnegie Mellon University, while colleagues at Microsoft Research strive to improve the head tracker. We intend to publicly release our software with a replaceable vision module, which will let vision research groups try their own approaches using our system. While waiting for computer vision to come up to speed, we might work with infrared-based vision systems so that we can push ahead with research on our virtual 3D conferencing environment.

Existing videoconferencing systems without gaze awareness present images that generally lack valuable social information. Such video streams quickly become uninteresting and are often ignored. In contrast, it's almost impossible to ignore eye contact. We believe that video that supports gaze awareness—along with low-latency, high-quality audio and easy call set up—will mean that a videoconferencing system finally has a chance to succeed. **MM**

Acknowledgments

Our work has benefited from insightful conversations with Jim Gray.

References

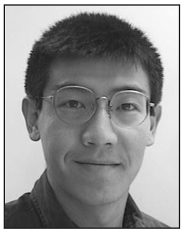
1. M. Argyle, *Bodily Communication*, International Universities Press, Madison, Conn., 1988.
2. D.G. Novick, D.G. Hansen, and K. Ward, "Coordinating Turn-Taking with Gaze," *Proc. Int'l Conf. on Spoken Language Processing (ICSLP'96),* Alfred I. duPont Institute, Wilmington, Del., 1996, pp. 188-191.
3. B. O'Connaill and S. Whittaker, "Characterizing, Predicting and Measuring Video-Mediated Communication: A Conversational Approach," *Video-Mediated Communication*, K. Finn, A. Sellen, and S. Wilbur, eds., Lawrence Erlbaum Associates, Hillsdale, N.J., 1997.
4. A. Colburn, M. Cohen, and S. Drucker, "The Role of Eye Gaze in Avatar Mediated Conversational

Interfaces," Tech Report MSR-TR-2000-81, Microsoft Research, Redmond, Wash., July 2000.

5. S.M. Anstis, J.W. Mayhew, and T. Morley, "The Perception of Where a Face or Television 'Portrait' Is Looking," *American J. of Psychology*, Vol. 82, No. 4, 1969, pp. 474-489.
6. B. Rime and L. McCusker, "Visual Behaviour In Social-interaction—Validity Of Eye-Contact Assessments Under Different Conditions of Observation," *British J. of Psychology*, Vol. 67, No. 5, 1976, pp. 507-514.
7. A.M. Noll, "Effects of Visible Eye and Head Turn on Perception of Being Looked At," *American J. of Psychology*, Vol. 89, No. 4, 1976, pp. 631-644.
8. C. Zitnick, J. Gemmell, and K. Toyama, *Manipulation of Video Eye Gaze and Head Orientation for Video Teleconferencing*, Microsoft Research Tech Report, MSR-TR-99-46, Redmond, Wash., June 1999.
9. Z. Liu et al., "Rapid Modelling of Animated Faces From Video," Microsoft Research Tech Report, MSR-TR-2000-11, Redmond, Wash., Feb. 2000.
10. W.H. Press et al., *Numerical Recipes in* C, second ed., Cambridge University Press, New York, 1992, pp. 683-688.
11. K. Toyama and G. Hager, "Incremental Focus of Attention for Robust Vision-Based Tracking," *Int'l J. of Computer Vision*, Vol. 35, No. 1, 1999, pp. 45-63.
12. A. Blake and M. Isard, *Active Contours*, Springer-Verlag, New York, 1998.
13. T. Kang, J. Gemmell, and K. Toyama, *A Warp-Based Feature Tracker*, Microsoft Research Tech Report, MSR-TR-99-80, Redmond, Wash., October, 1999.

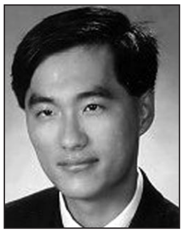
Jim Gemmell is a researcher in the Microsoft Research Telepresence Research Group at the Bay Area Research Center, San Francisco. His research interests include telepresence, digital audio and video storage/retrieval, and reliable multicast. He produced the online version of the ACM'97 conference. He received his PhD and BSc from Simon Fraser University and his M.Math from University of Waterloo.

Readers may contact Gemmell at Microsoft Research, 301 Howard St., Suite 830, San Francisco, CA 94105, e-mail jgemmell@microsoft.com.



Kentaro Toyama is a researcher at Microsoft Research in Redmond, Washington. His primary research interest is in vision-based tracking as applied to human-computer interfaces, video teleconferencing, graphical avatar animation, and video understanding. He received an AB in physics from Harvard University in 1991 and a PhD in computer science from Yale University in 1998.

C. Lawrence Zitnick is pursuing his PhD in robotics at the Robotics Institute, Carnegie Mellon University. His research interests include stereo vision, pattern recognition using neural and belief networks, and telepresence. He received a BS in mathematics and computer science from Carnegie Mellon University in 1996, and was awarded the Allen Newell Award for Undergraduate Research in 1995.



Thomas Kang is a graduate student in computer science at Carnegie Mellon University. He has worked at Schlumberger Austin Research and Microsoft Research on various projects ranging from sonification to video conferencing. His current research focuses on video analysis and compositing. He received a BA in computer science from Harvard University in 1996.

Steven Seitz is an assistant professor in the Department of Computer Science and Engineering at the University of Washington, where he conducts research in computer graphics and computer vision. His current research focuses on the problem of acquiring and manipulating visual representations of real environments. He received his BA in computer science and mathematics at the University of California, Berkeley in 1991 and his PhD in computer sciences at the University of Wisconsin, Madison in 1997. He was corecipient of the 1999 David Marr Prize for best paper at the International Conference of Computer Vision, and received an NSF Career Award in 2000.