Perceptual requirements and proposals for the UbiCom augmented reality display

W. Pasman, 1997

Introduction

In this report I will propose some perceptual requirements for the Augmented Reality (*AR*) display for the first prototype of the UbiCom project. It would be appropriate to define the task to be done with our AR system first (Pasman, 1997; Davis et al. ,1994; Ellis, 1997), but the UbiCom project is mainly a technology-driven project, and therefore there is no single final task that the system has to support. Rather a number of very general tasks should be feasable to run: maintenance and control, route planning, safety/inspection, acquiring information from Internet, building complex (spatial) constructions, playing games such as an adventure, paintball, etc.

Consequently, we will have to be able to reach the highest performance levels required for each of these tasks. For example, for route planning it would be no problem if the arrow indicating the direction to go would not be aligned perfectly to the world and would disappear when it moved towards the periphery of the visual field. However, for a paintball game it would be annoying if the virtual container I'm hiding behind would be transparent or even invisible for other players.

In the first paragraph I will start with an overview of the sources of optical distortion. Next, perceptual consequences of distortions will be discussed, followed by a list of requirements. Thereafter I wil propose some solutions in an attempt to fulfill the requirements, and then some feasability estimations of these solutions will be made. Conclusions will be drawn how the UbiCom first prototype might look like.

Sources of optical distortion

There are a large number of optical depth cues (see Sedgwick, 1986 or Wickens, 1990 for an overview). The most important cues defining the depth impression of the observer, his *perceived depth*, are perspective, texture gradients, occlusion, movement parallax, shadows and binocular parallax/convergence. Figures 1 to 6 give some impression of what may happen when these cues are rendered erroneously in an AR system.

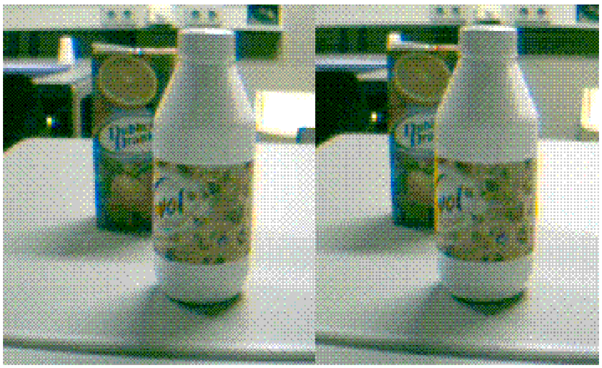


Figure 1. Stereoscopic image pair of a scene, without errors. Scene can be viewed by looking with the left eye to the left image and with the right eye to the right image.

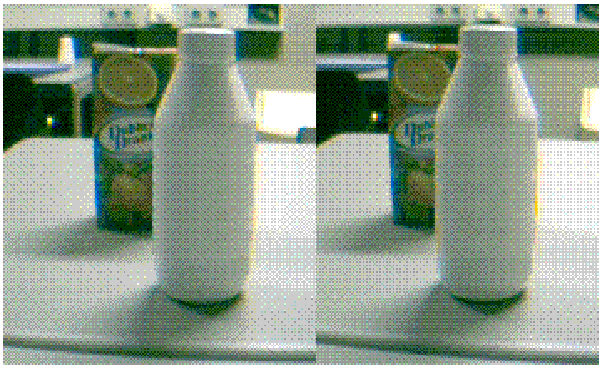


Figure 2. A virtual bottle without texture seems slightly flatter than the one without texture.

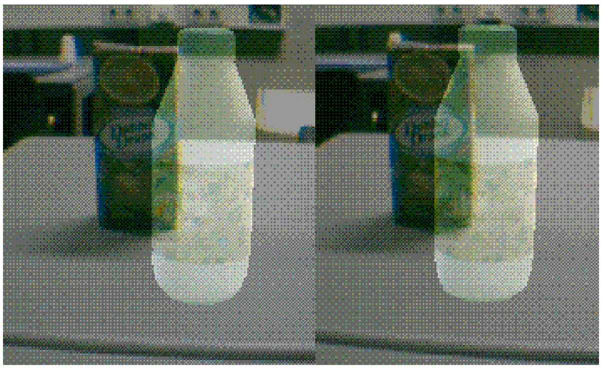


Figure 3. Absence of occlusion looks not very well. Note the darkened background in order to improve visibility of the virtual object.

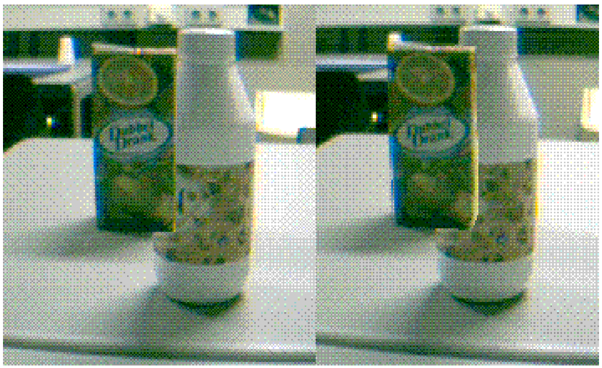


Figure 4. Failure to render occlusion correctly (here the real juice container was put in front of the virtual bottle) is extremely disturbing.

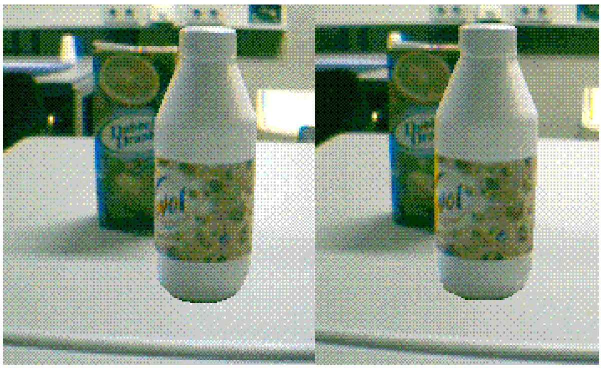


Figure 5. Without shadow the virtual bottle seems to float above table. Importance of shadow cues often are underestimated.

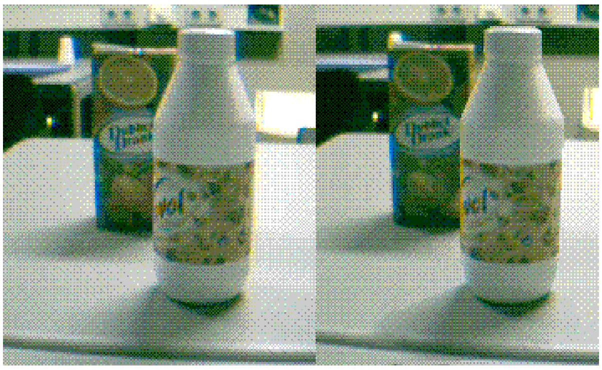


Figure 6. Incorrect binocular disparity, for example due to incorrect interpupillary distance, hardly influences the apparent depth.

Holloway (1997) enumerated the sources that cause a mismatch between the displayed image (that was made for a certain viewpoint) and the actual viewpoint of a non-moving observer (Table 1). The effect of such an *incorrect alignment* is illustrated in Figure 7. The error values give the approximate displacement of the virtual world from its correct position in millimeters. Figure 8 illustrates optical distortion caused by lenses, while Figure 6 showed the effects of an incorrect interpuppilary distance.

Table 1. Error sources and typical associated error caused by incorrect alignment (from Holloway, 1997).

	Error source	Error (mm)	Assumes
1	Lag between a movement and the corresponding update of the display	20-60+	Lag=20-180ms movement 500mm/s or 50˚/s
2	Optical distortion caused by lenses	0-20	display=HMD
3	errors in the placement of the the tracker in the real world	4-10+	
4	tracker measurement errors (static, dynamic, jitter)	1-7+	typical magnetic tracker at 50cm distance
5	acquisition/placing error, e.g. when objects were scanned with MRI	1-3	typical MRI voxel set 3mm
6	Mismatch of assumed and actual viewpoint	0-2+	virtual image at 50cm, 5mm eye movement, viewed point is 20cm from image plane
7	display grid deviates from perfectly rectangular	1-2	
8	Image translation, incorrect interpuppilary distance, aliasing	<1	good calibration, 640x480 resolution

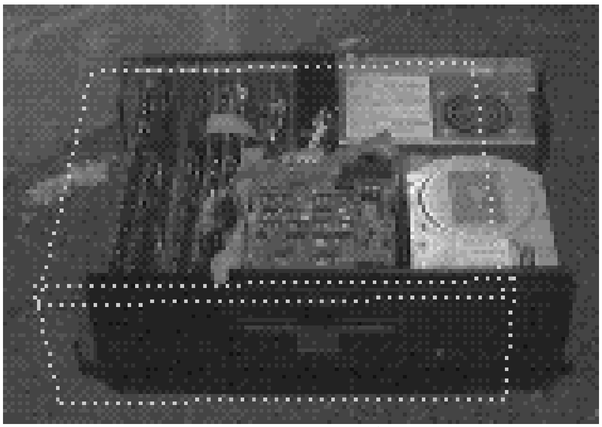


Figure 7. Example to illustrate the effect of an alignment error. The dashed lines represent the virtual computer box, and it is misaligned with the real box (from Uenohara, 1997).

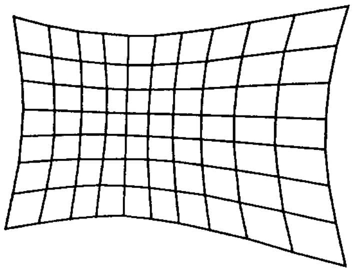


Figure 8. Typical lens distortion: pincushion (From Holloway, 1997).

Note that the error values of Table 1 are tricky: they indicate the amount of error that would be expected following the rules of geometry, but *humans are not expected to use these rules*. But there is evidence that the rules of geometry are quite a good predictor of perceptual distortion (Pasman, 1997), and therefore I will use geometry to estimate perceptual distortions. Furthermore, I would suggest to express errors in terms of visual angle (degrees) instead of millimeters displacement.

Perceptual consequences of distortion

Table 1 showed that lag is the most important source of geometric inequivalence in AR displays. Optical distortion from the lens system is less important, followed the accuracy of the viewpoint measurement. I will discuss the consequences of these issues in more detail now.

Alignment errors

As was already indicated by Table 1, positional inaccuracies are reflected directly into a geometric inequivalence. For orientation inaccuracies, the effect is even worse as the geometric inequivalence will grow linearly with the distance between the virtual object and the observer. Azuma (1997a,b) indicated that errors of a few pixels are noticeable, and that angular precision should be below 0.5°. In immersive VR the alignment of the real world with the virtual world is usually not very important; the immersed observer has very little references indicating a mismatch. In augmented reality this match will become crucial. For example for medical purposes an misalignment of a millimeter may already be fatal (Bajura, Fuchs & Ohbuchi, 1992), although for most tasks less precise aligments seems acceptable. Dynamic properties of the tracking will be discussed in the next paragraph.

Lag

Lag is a lag between the movements of the observer and the associated update of the virtual image. Lag causes an apparent displacement of the virtual objects relative to the real objects when the object or observer moves. So lag causes a misalignment, but only if the observer or object moves. Therefore, the perceptual effect is similar to the misalignment errors (Figure 7). In contrast with VR, with AR the effect of lag is directly visible as the observer has the real world as a reference. Lags smaller than 300ms are essential to uphold the sense of reality of the displayed objects (Wloka, 1995). For text-augmented real-world views, refreshing the text about 10 times per second (implying a lag of at least 100ms) seems sufficient (Feiner, MacIntyre, Haupt and Solomon, 1993). Both Feiner, MacIntyre, Höllerer and Webster (1997) and Mann (1997) already built a wireless, wearable AR display using text augmention.

But in many situations, especially those concerning virtual objects, lag is more critical. Even with a moderate lag of 50ms virtual objects appear to 'swim' through the real world (Azuma and Bishop, 1994). Padmos and Milders (1992) indicate that for driving simulations and helicopter simulations in immersive reality (where the observer can not see the normal world), lags should be below 40ms. For augmented reality the constraints will be even stricter. They suggest that the displacement of objects between two frames should not exceed the 15 arcmin (0.25°), which would require a maximal lag of 5ms even if the observer rotates his head with a moderate speed of 50°/s. Several other authors use a similar approach (Azuma and Bishop, 1994; Azuma, 1997a,b; Olano, Cohen, Mine and Bishop, 1995; Holloway, 1997) and come to similar maximal lags. Actually, during typical head motions speeds of up to 370°/s can occur (List, 1983), but I don't expect that observers rotating their head that fast will notice slight object displacements. Many authors suggest that 10ms will be acceptable for AR (Azuma and Bishop, 1994; Ellis, 1997; de Poot, 1995).

Several experiments have been done to estimate the effect of lag on observer performance. Ellis (1997) showed, for guiding a ring over a bent wire, that latency has a direct effect on the number of collisions of the ring with the wire, even with lags as small as 10ms. Keran, Smith, Koehler and Mathison (1994) showed, for tracking an oscillating target, that the average tracking error at 30ms lag is significantly less than that observed at higher lags. De Poot (1995) built a system with a lag of 8ms, which he found to be short enough to go for 'unlaged'. He showed that, for stationary objects, most observers can distinguish between lags of 8 versus 16 ms, and for an angular-velocity discrimination task he showed that 100ms lag has detoriating effects on the active motion perception. It can be expected that even lags of 8ms can be distinguished from lags of 0ms.

Occlusion errors

Occlusion is a dominant depth cue in many situations (eg, Ono, Rogers, Ohmo & Ono, 1988; Braunstein, 1982; Wickens, 1990), although occlusion is less dominant for text and wireframe-like structures (eg., Feiner, MacIntyre, Haupt and Solomon, 1993; Ellis and Menges, 1997; Feiner, 1995). Occlusion can dominate several other depth cues, such as motion parallax and stereoscopic cues (see below, under Stereoscopic images). Especially for games, it seems important that occlusion cues are handled correctly, for example one should be able to hide equally effective behind a virtual wall as if it were a real wall. Real tasks also are hindered by incorrect occlusion: for example Bajura, Fuchs and Ohbuchi (1992) noticed for ultrasound images projected over the body of the patient that the images looked as if they were painted on the body. To solve this, they projected a complete pit over the body (Figure 9).

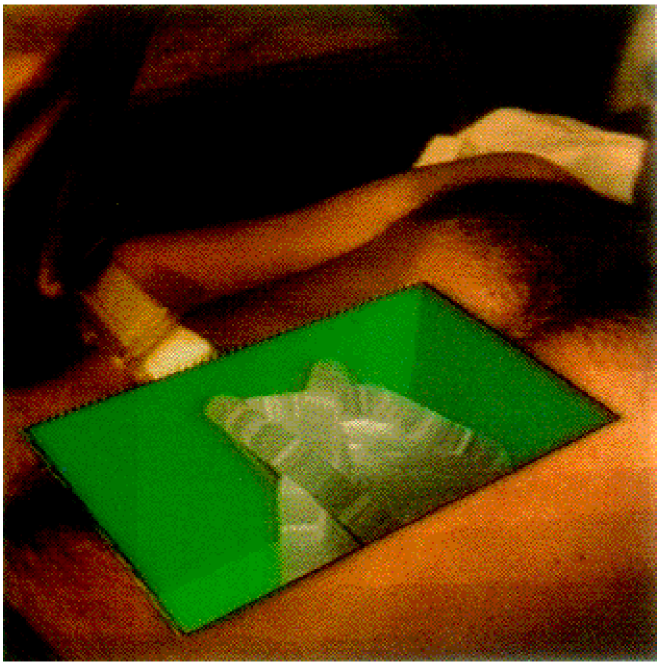


Figure 9

without the pit around the virtual stomach, the stomach would seem to be painted on top of the body (From Bajura, Fuchs and Ohbuchi, 1992).

Most AR systems currently do *not* provide the observer with appropriate depth cues. There seem to be two causes for this:

1. Current AR systems where the computer image and real-world image are merged optically (*optical AR*) use a normal display (LCD or CRT) to generate the computer image, and merge this image with the real-world image by means of a half-silvered mirror. Rendering occlusion of non-transparent object correctly in such a system would require a separate optical filter (eg., an black and white LCD) to block out parts of the real-world image. In systems where the images are merged electronically (video AR) the real world can be blocked easily. However, such systems are less popular as it requires the real world to be sampled, and this strongly reduces the quality of the real-world image (both temporarily and with respect to the resolution and contrast).

2. Rendering the occlusion cues correctly would require a depth map of the real-world image, and such a spatial map is difficult to obtain and maintain. For example, assume that the observer is looking at a virtual trash bin on the other side of the street, and a car is passing between the observer and the bin. The car thus would occlude the trash bin. But if the AR system is not able to track the car correctly, it seems impossible to render the -probably partially- occluded bin correctly.

Stereoscopic images

Stereoscopic images are image pairs, one for the left and one for the right eye. The viewpoints of the eyes are different, and therefore the differences in these image pairs contain depth information. A huge amount of work has been done comparing monocular and binocular displays (see Serdick, Davis, King and Hodges, 1997, for an overview), so I will discuss a small selection to come to a conclusion.

Supplying the augmented images to only one eye of the observer allows him to see the unaugmented image with the other eye. For the observer, this may look as if objects are transparent, and very disturbing perceptual artefacts may occur. As discussed under 'Occlusion', this is unacceptable for a number of applications. Therefore, the images should be provided to both eyes.

It is possible to provide exactly the same augmented image to both eyes. Monoscopic images would halve the amount of computational resources needed. In such a case the stereoscopic cues in the virtual image will be in conflict with the stereoscopic cues from the environment, but such a conflict probably has only small effects on the perception, for most tasks at least. Ellis (1997) suggested that stereoscopic cues mainly serve a 'camouflage-breaking' purpose. Occlusion cues will override stereoscopic cues (Wickens, 1990), just as occlusion will override motion parallax cues in case of conflict with occlusion cues (Ono, Rogers, Ohmo & Ono, 1988). Binocular disparity may be not effective at distances larger than 2m (Serdick, Davis, King and Hodges, 1997). Reinhart, Beaton and Snyder (1990) compared the effectivity of perspective, disparity, occlusion and luminance cues for a depth-comparison task at a distance of 2 meters, and found that disparity has no effect, neither on the apparent depth nor on the response time. Ehrlich (1994) found no advantages of stereoscopic images for tasks as object tracking, self movement, object manipulation en distance estimation. I expect that stereoscopic cues are of importance only if the other depth cues, especially occlusion, fails (Pasman, 1997; see also Cole, Merritt, Fore and Lester, 1990 or Spain & Holzhausen, 1991). In normally shaded objects stereoscopic depth there may be too little sharp contours that can be used for stereo matching (McWhorter, Hodges, & Rodriguez, 1991).

Clapp (1986) suggested that binocular disparity works fast (only auditory signals and the equilibratory senses are processed faster) and precise (in the order of 0.1°).

Paradoxically, it is often found that human observers judge disparity as the most important and compelling depth cue (eg, Ono, Rogers, Ohmo & Ono, 1988). Especially for games, such a 'compellingness' is required. Furthermore, a disparity error also may cause alignment errors. I would suggest some 'quick and dirty' approach to stereoscopy: having stereoscopic cues about right is sufficient.

Accomodation - convergence coupling

As the observer focuses his eye-lens to get a sharp image from the object he is looking at, this focusing gives him distance information. This depth information is called the 'accomodation cue'. Accomodation cues are largest for close objects, with an accuracy of 1/4 inch according to Clapp (1986). Ellis and Menges (1997) showed that accomodation cues are effective when an image is provided to only one eye, but much less when both eyes get an image. Gooding, Miller, Moore and Kim (1991) found a scaling of the perceived depth over the geometric depth (perceived/real depth) of 65% at a distance of 67 cm up to 85% at 268cm, and suggested that these effects were due to the accomodation cues.

Finally, there exist strong couplings between accomodation of the eye lens and convergence of the two eyes. Mismatches between accomodation and convergence cues are known to cause visual fatigue (Ellis and Adelstein, 1997). I am not sure whether the convergence regulates the accomodation or the other way round, or even in both directions.

List of requirements

I propose to aim at a lag of at most 10ms, a reasonable value according to most researchers. A stereoscopic color display is preferable, and it should minimise visual fatigue due to accomodation/convergence conflicts. The mixing of the real virtual world should be done optically. I expect that dimming the light from the environment in order to enhance the visibility of the AR display (sunglasses-effect) is acceptable. If the interpuppilary distance is adjustable, this distance should be supplied to the rendering machinery in order to generate the geometrically correct images. If possible, occlusion should be rendered correctly.

The field of view is a difficult point. As suggested above, completely blocking out parts of the world, in order to render occlusion correctly, is important for several game-like situations. But for most applications, blocking out only in the center of vision seems sufficient. I expect that complete blocking will rise many technical problems that may be better to avoid in a first attempt to build an AR system. For example, with a large field of view high update rates are required to prevent flicker (Padmos and Milders, 1992). Furthermore the display may need to be curved, which seems difficult from optical point of view. Furthermore, distortions in the peripheral image will have different perceptual consequences than distortions in the centre of vision. On the other hand, average eye movements are within the range of about ±30° and taking a field of view smaller than 60° therefore may hinder exploration of the virtual environment. I would propose to use a horizontal field of view of 60-80° for each eye.

Possible solutions

This section will discuss possible solutions for problems due to lag, occlusion, convergence/accomodation, stereoscopic images and tracking problems.

Lag

Holloway (1997) enumerated the sources of lag and estimated typical values associated with each type of lag (Table 2). I will discuss those types of lag below.

Table 2. Sources of lag (after Holloway, 1997)

Lag type	Description	Typical lag
Tracker lag	internal computations inside the tracker	5 ms
Host-computer lag	transmitting tracker data, background OS tasks	10 ms
Image-generation lag	rendering time	25 ms
Video sync lag	lag while waiting for next video frame	10 ms*
Frame lag	scantime from start of display to a certain raster position	10 ms*
Internal display lag	some LCD displays wait until a full frame has been received before displaying it; others refresh only part of the pixels during one refresh cycle	10ms

A. Tracker lag

I will discuss tracker lag below in the next subsection, 'Tracking'

B. Host-computer lag

Jacoby, Adelstein and Ellis (1996) showed that for magnetic trackers transmission lags can be minimised to Xms by using special drivers and fast, non-serial interfacing. For some purposes, fast A/D converters can be used to transfer the viewpoint to the rendering computer (eg., de Poot, 1995).

Furthermore, it is essential to use a low-overhead or perhaps even guaranteed-latency operating system for all time-critical paths in our system. For example Unix seems inappropriate, as no maximal-time guarantee is given for any action, often breaks of about 200ms? occur (Azuma, 1997) and serial ports are polled at a low rate (Cohen & Olano, 1995). Similarly, on many PCs spontaneous actions of several hundreds of milliseconds can occur, even in MS-DOS mode.

C. Image generation lag

We can distinguish four types of image generation (*rendering*): polygon rendering, ray tracing, image morphing and light field rendering. Polygon rendering basically projects the corners of polygons to the display, after which the corners are connected with straight lines; next the polygons are filled with color or texture. With ray tracing, the light rays coming from the scene to the eye of the observer are traced back: from the eye through the display until an object in the virtual scene has been reached. The texture and color of the point where the ray hits the object is determined, and this color is assigned to the pixel where the ray went through the display. With image morphing, parts of an existing image are deformed, moved around etc., in order to match a new viewpoint. With light field rendering, the basis of the rendering is a 4-dimensional array containing all possible light rays travelling through a scene (given some lighting). Arbitrary images can be extracted from such an array (Levoy & Hanrahan, 1996).

Here I will not consider raytracing and light-field rendering, as there exist no machines capable of real-time rendering using these techniques. The hardware for polygon-rendering is continually getting faster. Currently, 15 to 90 million textured pixels can be rendered per second with standard PC cards (eg., see Quantum3D, 1997), which was once

* Half the field-time, which is 20ms for a 50Hz display

a state-of-the art performance (Molnar, Eyles and Poulton, 1992). Current high-end machines can render 710Mpixels/s (Montrym, Baum, Dignam & Migdal, 1997).

So far, there seems to be little problems with the rendering speed. However, the number of polygons often are excessive. Therefore, a number of alternative approaches have been proposed to avoid transforming and rerendering of all polygons for each frame (Table 3).

Table 3. Ways to lower rendering time.

Working principle	References
Faster (polygon) rendering	Quantum3D, 1997; Montrym, Baum, Dignam & Migdal, 1997; Molnar, Eyles & Poulton, 1992
Lower the number of polygons in the scene	Michel and Brock, 1997; Hoppe, 1996; Certain, Popovic, DeRose, Duchamp, Salesin and Stuetzle, 1996;
Reuse of parts of earlier renderings	Shade, Lischinski, Salesin, DeRose and Snyder, 1996; Mann end Cohen-Or, 1997; Torborg & Kajiya, 1996
Render distant parts and closer parts separately, each with an appropriate update rate.	Shade, Lischinski, Salesin, DeRose and Snyder, 1996
Reuse parts of earlier polygon transformations	?
Rerender random pixels in image instead of full image	Bishop, Fuchs, McMillan & Zagier, 1994

Rerendering of random pixels seems suited only to ray-tracing, a technique that seems too slow to use for AR purposes. Image morphing, or more generally image-based rendering, gets much attention at the moment; even Microsoft is seriously working to implement hardware supporting image-based rendering (Seitz and Dyer, 1996; Torborg & Kajiya, 1996; Cohen, Levoy, Malik, McMillan & Chen, 1997). Reusing earlier renderings can be done in several ways. The most straightforward way is to 'morph' the previous rendering and to rerender/correct the most disturbed parts in the morphed rendering. A more sophistocated way is to render objects separately in order to have separate images of different objects, allowing to morph them separately. The following two sections will discuss these approaches and their problems.

C1: reusing complete images

Several ways exist to generate new views from earlier images off-line (Havaldar, Lee and Medioni, 1997; Debevec, 1996; Horry, Anjyo & Arai, 1997), but these approaches reconstruct the 3D layout of the scene and the textures associated with the reconstructed polygons, using standard polygon rendering techniques to generate new views, and therefore are not really 'image morphing' tools (Debevec, Taylor & Malik, 1996; 3D Builder, 1996). Furthermore, reconstruction of 3D layout still requires human intervention and cannot be done fully automatically. But these techniques show that the number of polygons can be reduced drastically while maintaining realistic images.

But morphing images in real-time thus requires different approaches. The general idea to use morphing to decrease lag is to display morphed versions of earlier images while waiting for the next completely rendered image from the polygon renderer (Mann and Cohen-Or, 1997).

There are some typical problems when this is done in a straightforward way (Mann and Cohen-Or, 1997). When such morphing actions are done given the previous image, information will be lacking to fill some parts of the image (the *visibility gap*). Figure 10 illustrates such a visibility gap and the way this is solved by Mann and Cohen-Or (1997). The growing and sudden collapse of these gaps is visually very disturbing, especially since the observer is exactly looking at the disoccluding side of objects as there may appear new objects exactly there. For augmented reality this problem may be less severe, as there will be, on the average, less objects behind each other than in immersive VR. Second, it is not clear whether this approach is suited for other simulations than walkthroughs (see Cohen, Levoy, Malik, McMillan and Chen, 1997). Third, there are artefacts caused by this morphing, that are illustrated in Figure 11. Fourth, there the previous image will have to be larger than the display, to be able to cope with rotational head movements of the observer. A way to solve this problem may be by using spherical images, such as Quicktime VR (Apple, 1997; Heid, 1996; Szeliski and Shum, 1997), but I don't know of combined morphing/Quicktime VR approaches.

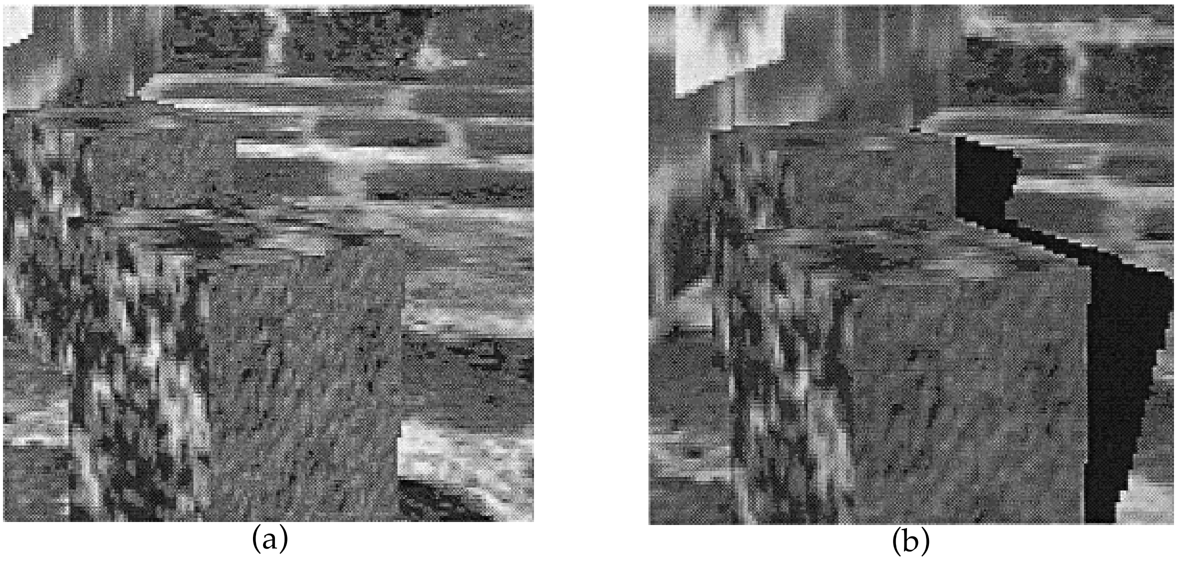


Figure 10a,b. Left an unmorphed scene. Right a morphed image such that the scene is seen more from the right. The black holes are the visibility gap: a now disoccluded part that was not visible in the unmorphed scene. (From Mann and Cohen-Or, 1997).

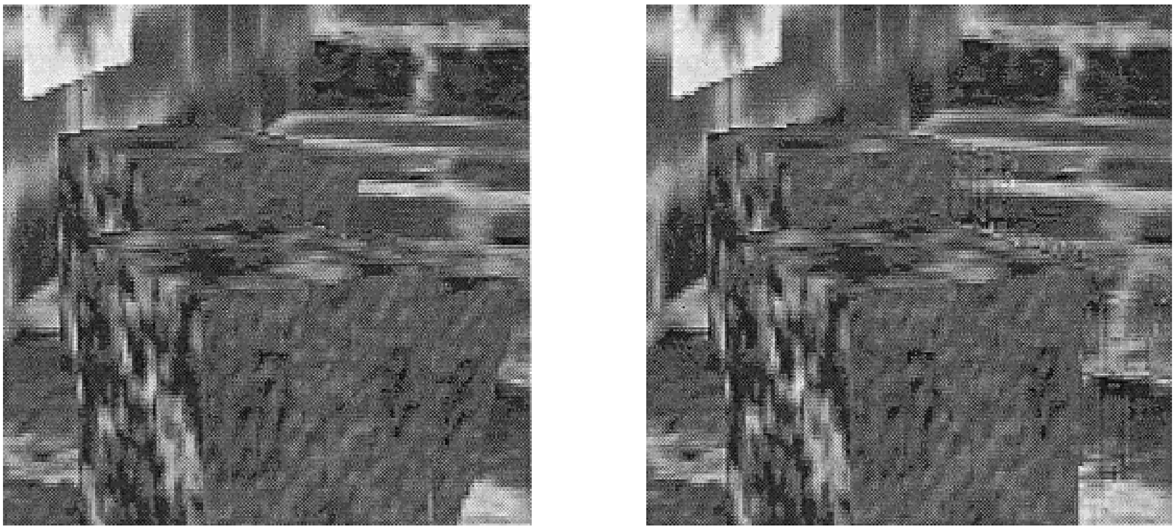
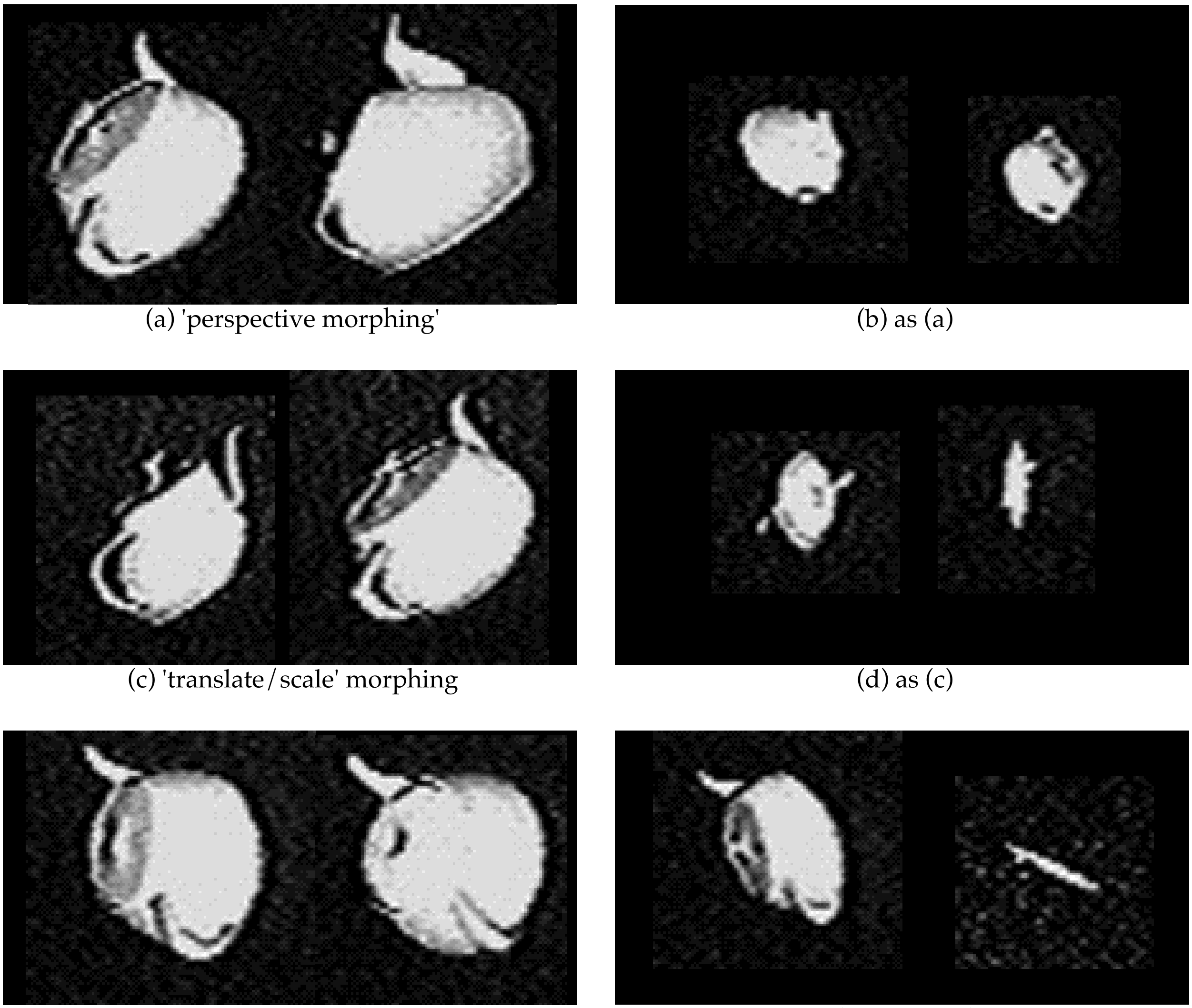


Figure 10c,d. Left, the visibility hole is filed with the extrapolated foreground object. Right shows the new view after correcting data has been generated. Note the waves-artefact where the visibility hole was, which may be caused by the lossy compression used to transmit the correcting data (From Mann and Cohen-Or, 1997).

C2: reusing partial images

By rendering parts of the scene separately, the problems with the visibility gap can be avoided. Version 3 of Apple's Quicktime VR (Apple, 1997) adds sprites to their omni-viewer. However, this combination allows only limited, precalculated correction of the sprite but no full control over the perspective. Microsofts Talisman architecture (Torborg & Kajiya, 1996; Lengyel & Snyder, 1997) offers better possibilities for this, as their architecture combines image morphing with polygon rendering. Horvitz and Lengyel (1997) proposed strategies to minimize distortions with this architecture. However, I expect that this architecture is worse in coping with head rotations than the Quicktime VR approach, as the Talisman system seems incapable to use images surrounding the observer.

The research that has been done on the Talisman architecture shows some of the limits of image morphing . To illustrate the typical *morphing artefacts* associated with too much reliance on morphing, some snapshots from a video of the Talisman system (Lengyel & Snyder, 1997) are shown in Figure VV. These videos suggest that an update rate of 13% is just acceptable (meaning that each 8 frames the image is replaced by a fresh polygon rendering), and 25% gives quite good results. These figures are for a quite fast-moving and rotating tea pot, for slower moving objects lower and finer interpolation even lower update rates seem acceptable.



(a) 'perspective morphing'

(b) as (a)

(c) 'translate/scale' morphing

(d) as (c)

(e) 'affine' morphing

(f) 'shade' morphing

Figure 11. Screensnaps of teapot-images morphed to simulate rotation. The teapot should rotate with about 7.5° in each step, but the source-image was re-rendered from a famous polygon model only once each 16 steps. This gives distortion which is extreme just before the image is updated. The image pairs shown here are screen-snapshots with extreme distortion just before re-rendering and the (correct) image just after a re-rendering. The first row shows a morphing strategy labelled 'perspective', the second row 'translate/scale', the third row 'affine' and 'shade'. The 'affine' morphing gives the least distorted images in most of the cases. (From Lengyel & Snyder, 1997).

D: Video sync lag

High rendering speeds do not necessarily imply low lags, as much depends on the pipelining of the rendering machine (Olano, Cohen, Mine & Bishop, 1995). For example, Jacoby, Adelstein, and Ellis (1996) optimized their Silicon Graphics configuration with a dual pipeline reality engine and a polhemus fastrak, and got a lag of 30ms. The optimized system of Keran, Smith, Koehler and Mathison (1994) had the same lag. But some other rendering approaches (Table 4) give also possibilities to reduce the lag.

Table 4. Ideas to decrease lag

Name of idea	Reference	Working principle
SLATS, Pixel-Planes 5	Olano, Cohen, Mine & Bishop, 1995	Careful pipelining by parallelizing transformation, rendering and frame buffering
Visual Display Research Tool	Burbidge & Murray, 1989	image transrotation just before displaying
Address recalculation pipeline	Regan & Pose, 1994	image transrotation just before displaying
'Image morphing' (IS DIT HARDWARE?)	McMillan & Bishop, 1995; Mann & Cohen-Or, 1997	morph image just before displaying
Just-in-time pixels	Olano, Cohen, Mine and Bishop, 1995	during rendering: per-pixel (or frameline) anticipation of the lag till displaying time
'Sprite morphing' Talisman	Torborg & Kajiya, 1996; Lengyel & Snyder, 1997	render objects separately and transform (rotate,translate,scale) these renderings to for intermediate 'renderings'

The pipelining-optimization approach of Olano, Cohen, Mine and Bishop (1995) can handle only small data sets (100 to 250 polygons). It is not clear whether these 200 polygons must have a limited size, but this problem seems to indicate that not the rendering of the polygons but the projection and clipping of the polygons is the real bottleneck in their setups. In spite of their effords the lag still is 16.7ms (on a 60Hz machine).

Translating and rotating the image just before displaying (Burbridge & Murray, 1989) can be done in hardware (Regan & Pose, 1994), which gives minimal lag for head rotations. These solutions require dedicated hardware next to a standard polygon renderer. A disadvantage of this approach is that only head rotations can be compensated, but not head translations and object movements.

Image morphing as proposed by Mann and Cohen-Or (1997) does not solve the video sync problem, as they propose to re-render the scene without textures in order to find the morphing settings. Such a strategy seems to make little sense as most current render hardware supports hardware texture mapping. Nevertheless such morphing can be implemented in hardware, if the z-value for each pixel in the image is known. I have not found such hardware in my literature search, but I expect that it exists and that it can reduce the lag drastically.

Just-in-time pixels (Olano, Cohen, Mine and Bishop, 1995) suggested to predict for each pixel, at rendering time, the position the camera will have when the pixel will be displayed. Their system predicted each line instead of each pixel; recomputing the whole geometry is very time-consuming and I am surprised that they can recompute 200 polygons each scanline. Furthermore I don't know any fast and efficient strategies to render a single line of a scene. In fact I wonder whether they actually used this strategy in their implementation, as they found a lag of 17ms which is exactly one frame (and an additional lag of 30ms of their tracker!).

Sprite morphing was already discussed above, under 'C2: reusing partial images'. This idea seems easily adaptable to decrease lags. For example the Talisman architecture (Torborg & Kajiya, 1996) is capable of real-time sprite morphing in hardware. This morphing is done just ahead of the raster beam of the monitor. Fresh morphing values are therefore displayed immediately. However, as noted above this architecture lacks a Quicktime-like improvement in order to cope with head rotations of the observer.

E. Frame lag

Because pixels on the display are usually shown sequentially instead of all at once, there is an additional lag of 40ms for the bottom-right pixel as compared to the top-left pixel (on a standard 50Hz TV display). Non-interlaced displays improve this to 20ms, but this is still twice as high as the total lag that we allow! Most solutions for the video sync lag (see previous paragraph) also can compensate for frame lag.

F. Internal display lag

Finally a problem that seems to have a simple solution: just use displays with minimal internal lag. If LCD displays are to be used, this point should be checked carefully.

Occlusion

In order to get the occlusion right, we probably need an up-to-date depth map of the real environment of the observer. GIS databases won't suffice, as they won't contain moving or moveable objects, such as room furnishment or cars moving through the street. I don't expect that such objects can be recognised, tracked and depth-estimated instantaneously with the video-camera's.

I propose to use dedicated hardware to construct depth maps in real-time based on the images from two cameras (Kano, 1995). The generated z-maps can easily be merged with computer-generated images by using the z-maps generated by the rendering hardware. An existing implementation is quoted to generate 30 z-maps per second for a 200x200 camera. This z-map also seems useful for other purposes, for example updating GIS-databases. I expect that this hardware will fail in some conditions and that in such a case some other mechanism is required to handle occlusion in a reasonable way.

If we want to avoid transparent, or ghostlike, virtual objects, we will need to block out the real-world image. Probably this can be done with a black and white LCD display, blocking parts of the real world where objects in the virtual world are closer to the viewer than the objects in the real world. A kind of optical device working as a video mixer would be preferable, but seems out of the scope of the UbiCom project.

Convergence/accomodation

With a retinal scanning display the accomodation of the eye has no effect on the image as projected on the retina. There will be no visual conflicts if accomodation is guided by convergence. However, the accomodation of the eye may also guide the convergence distance, and if this is the case the problem still exists, although in a different way. But I expect that with a retinal scanning display no large convergence-accomodation conflicts will occur.

We can expect that our first prototype system will have to use a normal, non-retinal scanning display. I propose that such a display should provide at least some way to adjust the accomodation distance, in order to prevent visual fatigue.

Stereoscopic images

Ellis (1997) showed that a conflict between stereoscopic cues from the virtual and real world is not necessarily problematic, as long as the disparity between both images is approximately correct for the observed objects. Therefore, displaying a single image to both eyes, and adapting the disparity of that single image quickly to the object the observer is looking at may be feasable. However, this will require the eye tracking and update of the displays to be done very fast, and I have no reference that this ever has been attempted to do this way. I propose to render each object separately to a single image, and to place the image in both views at the correct disparity distance.

Tracking

Fast and accurate tracking is still a problem with AR systems, and problems will complicate even further as the space the observer is allowed to move through is enlargened. Holloway gave precise accuracy measurements for the Polhemus Fastrak and the Flock of Birds (Holloway, 1997). Table 5 shows the relevant parameters for some current commercially available trackers (The Virtual Reality Source, 1997). A favourite tracker is the Polhemus Fastrak, especially due to its low lag. Jacoby, Adelstein and Ellis (1996) showed that the lag with the Polhemus Fastrak can be reduced to Xms using special drivers and a parallel port instead of the usual serial port. It is not clear whether the Polhemus accuracy is reached in practical setups (Cohen & Olano, 1995).

Table 5: properties of some commercially available trackers. Note that maximal lag = lag+1/refresh rate (from The Virtual Reality Source, 1997).

Tracker vendor/type	lag	refresh rate (Hz)	accuracy	range (m)	price (min)
Polhemus/Insidetrak	12ms	60	0.0003 "/" [**] ; 0.03°	1.5	$1,498
Polhemus/Fastrak	4ms	120	0.0002"/" [**]; 0.15°	10	$6,000
Polhemus/Ultratrak PRO	6ms	120	1"; 3°	5	$21,500
Polhemus/Startrak	NA	120	2"/3°	8	$62,500
Ascension/Flock of Birds	NA	144	0.07"; 0.5°	6?	$2,645
Ascension/Minibird	NA	144	0.07"; 0.5°	6?	$3,995
Ascension/Spacepad	NA	120	depends on range	upto 10?	$1,158
Ascension/Motion star	NA	144	~0.005 "/"[**] ~0.3°/m	12	$19,315
Ascension/Motion star wireless	NA	144	~0.005 "/"[**] ~0.3°/m	12	$55,815

A technique to attack the lags is by using a predictive filter (Feiner, MacIntyre, Haupt and Solomon, 1993; Azuma and Bishop, 1994, Lewis, 1986; Welch & Bishop, 1997), usually a Kalman filter (Kalman & Bucy, 1961). But these filters introduce other artefacts, such as overshooting, which will get disturbing when predicting over time spans larger than about 80ms (Azuma and Bishop, 1994). Magnetic tracking seems a good choice for a first prototype.

The global positioning system (*GPS*), which uses GPS-satellite signals to calculate the position of the GPS-receiver, has world-wide tracking capabilities, but at the expense of accuracy. I don't have data on the lag of GPS systems. Normally its accuracy is very low (typical errors of 100m), but an additional GPS-transmittor on the ground broadcasting error-correction information increases the accuracy to typically 1m. Nevertheless, this seems too rough for most AR applications. When the AR displays adds only text to the real-world image, deviations of some degrees may be acceptable, but for close objects and graphical overlays only a few tenths of a degree seem acceptable.

[**] inches of displacement error per inch of separation

Much work on optical tracking has been done by Azuma, who workes with optical markers mounted on the ceiling. Cameras on the helmet track those markers, and by doing inverse optics the current viewpoint can be calculated very accurately by using only N ..x.. cameras. In 1990 he suggested that a lag of 5ms with an accuracy of 2 mm and 0.1° should be possible (Wang, Azuma, Bishop, Chi, Eyles and Fuchs, 1990). The system of Ward, Azuma, Bennett, Gottschalk and Fuchs (1992) reached this accuracy, but had 20-60ms lag. More recently Azuma (1997) indicated that optical trackers still have a lag of 15-30 ms. Furthermore, placing markers seems impractical for outdoor use and quite expensive for indoor use.

Several authors (Vallino, 1997; Azuma & Bishop, 1994) have proposed to combine different techniques. I think that the approach of Azuma and Bishop (1994), combining optical tracking with inertial tracking, looks most promising. I expect that a combination of optical tracking using markers in the GIS database, fast precision-gyroscopes for orientation measurement and accelleration-sensors will meet our requirements. The gyroscope and accelleration-sensor (the inertial tracker) can be used to extrapolate the slow and delayed measurements. It may be possible to do the optical tracking on the backbone, depending on lags. Probably a GPS is not required if this combination shows stable. Updating tracker data directly after partial sensor data has been retrieved instead of waiting for complete sensor data (Welch & Bishop, 1997) also seems a promising way to attack the problems.

Feasability and costs of solutions

Reaching a lag of 10ms with off-the shelf components is quite hard. Using an off-the-shelf 3D card may be fast enough to render 300 frames per second (but most cards are specified only up to 200Hz). Two such cards would be needed to render a stereoscopic image. When this approach is used, other tricks to improve rendering speed become important, but given a refresh rate of 300Hz 90Mpixels/s would allow us to render 300,000 pixels in each frame (eg. a full frame of 600x500 pixels); polygon processing and erasing of the Z-buffer might take more time than the rendering itself. Special care should be taken to avoid latency due to pipelines in the rendering hardware (Cohen & Olano, 1994; Wloka, 1995; Azuma, 1997). It seems harder to find a display capable of a refresh rate of 300Hz, but possibly one exists. This combination would result in a lag of 6.7ms (assuming that no additional pipelining occurs within the rendering engine), leaving 3.3ms for tracking. A theoretic solution would be to use a display with fast refresh. Suppose that a display capable of refreshing all image pixels within 1ms, and then keeps displaying them until the next frame is available. This seems possible with current LCD panels. However, this only transforms the lag problem to another lag problem, in fact it seems better to have either an up-to-date image or no image at all.

The feasability of this first approach using off-the shelf components is dubious: I don't expect that 300Hz HMDs are available commercially. Maybe we can modify an existing CRT-based HMD, but I don't know whether CRT-color display-based HMDs exist and how they will behave when modified to 300Hz. Furthermore, the RSD is not ment for such high refresh rates either.

A more realistic approach is to minimize both image generation lag and video sync lag by using special hardware that composes the image just in front of the raster beam (I don't know whether this approach would make sense for LCD displays, but it does for the RSD). The image can be morphed given the z-values of each pixel, or alternatively the rendered pictures of the objects can be transformed. I think the per-object morphing is to be preferred, given the problems with the full-image warp. I would propose to render some Quicktime-like image as background and place separate sprites in the foreground; special hardware should be used to compose the image just ahead of the raster beam. Possibly the Talisman-hardware can be adapted to do this. I think that this approach has a reasonable feasibility, as both hardware-morphing techniques and hardware-Quicktime techniques (Regan & Pose, 1994) exist.

Conclusion

We should strive for a system with a maximal total lag of 10ms between movement of the observer and the corresponding update of the display. I propose to use hardware to compose morphed sprites over a Quicktime-VR like scene in real time, just ahead of the raster beam. This would allow us to respond quickly to observer movements (both rotations and translations) with minimal rendering capacity required. With such a combination, it may be possible to do the polygon rendering on the backbone, but this has to be investigaged and will deped on the communication- and rendering lags. Low-latency tracking may be done by combining optical tracking with inertion tracking.

References

3D builder [Computer software] (1996). 3D construction company, Elizabethton, Tennessee, USA. Apple 1997

Azuma, R. and G. Bishop (1994). Improving static and dynamic registration in an optical see-through hmd. Proceedings SIGGRAPH '94 : 197-204.

Azuma, R. T. (1997). A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments 6, 4, 355 - 385. Earlier version appeared in Course Notes #9: Developing Advanced Virtual Reality Applications, ACM SIGGRAPH (Los Angeles, CA, 6-11 August 1995), 20-1 to 20-38. Available Internet: www.cs.unc.edu/~azuma/ARpresence.pdf.

Azuma, R. T. (1997b). Registration errors in augmented reality. Available Internet: http://epsilon.cs.unc.edu/~azuma/azuma_AR.html.

Bajura, M., Fuchs, H., & Ohbuchi, R. (1992). Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. Computer graphics - Proceedings of the SIGGRAPH'92, 26 (2), 203-210.

Braunstein, M. L. (1982). The use of occlusion to resolve ambiguity in parallel projections. Perception & Psychophysics, 31, 261-267.Burbridge, Murray 1989

Clapp, R. E. (1986). Stereoscopic displays and the human dual visual system. SPIE 624: Advances in display technology VI, 41-52.

Cohen, J., & Olano, M. (1994). Low latency rendering on pixel-planes 5. UNC Chapel Hill department of computer science technical report TR94-028. Available FTP: //ftp.cs.unc.edu/pub/users/cohenj/LowLatency/ lowlat.ps.Z.

Cohen, M., Levoy, M. (1997), Malik, J., McMillan, L., & Chen, E. (1997). Image-based rendering: Really new of déjà-vu?. Proceedings of the SIGGRAPH'97 (Los Angeles, CA, August 3-8), 468-470.

Cole, R. E., Merritt, J. O., Fore, S., & Lester, P. (1990). Remote manipulator task impossible without stereo TV. *Proceedings of the SPIE, 1256*, 255-265. Bellingham, WA: SPIE.

Davis, E. T., Wickens, C. D., Barfield, W., Ellis, S. R., Ribarsky, B., Corso, G. M., & Eggleston, R. G. (1994). Human perception and performance in 3D virtual environments. Proceedings of the human factors and ergonomics society 38th annual meeting, p. 230-234.

Poot, H. J. G. de (1995). *Monocular perception of motion in depth.* Unpublished doctoral dissertation, Faculty of Biology, University of Utrecht, Utrecht, The Netherlands.

Debevec, P. E. (1996). Modelling and rendering architecture from photographs: A hybrid geometry- and image-based approach. Computer graphics proceedings (SIGGRAPH'96), 11-20.

Debevec, P. E., Taylor, C. J., & Malik, J. (1996). Modelling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In proceedings of the SIGGRAPH'96, 11-20.

Ehrlich, J. A. (1994). Are stereoscopic displays beneficial in virtual environments? *Proceedings of the human factors and ergonomics society 38th annual meeting*, 952.

Ellis, S. R., & Adelstein, B. D. (1997). Visual performance and fatigue in see-through head-mounted displays. Available Internet: http://duchamp.arc.nasa.gov/research/seethru_summary.html.

Ellis, S. R. (1997b). Virtual Reality: success or failure?. Proceedings of the CSN'97 (Utrecht, The Netherlands: 25 November).

Ellis, S. R., & Menges, B. M. (1997). Judgments of the distance to nearby virtual objects: Interaction of viewing conditions and accomodative demand. Presence, 6 (4), 452-460.

Feiner, S. (1995). KARMA. University of Columbia, Department of Computer Science. Available Internet: http://www.cs.columbia.edu/graphics/projects/karma.

Feiner, S., MacIntyre, B., Haupt, M., & Solomon, E. (1993). Windows on the World: 2D Windows for 3D Augmented Reality. Proceedings of ACM Symposium on User Interface Software and Technology (Atlanta, GA, November 3-5), 145-155. Available Internet: www.cs.columbia.edu/graphics/projects.

Feiner, S., MacIntyre, B., Höllerer, T., & Webster, A. (1997). A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. Proceedings ISWC'97 (International Symposium on wearable computing, Cambridge, MA, October 13-14), 1997. Available FTP: www.cs.columbia.edu/graphics/publications/ISWC97.ps.gz.

Gooding, L., Miller, M. E., Moore, J., & Kim, S. (1991). The effect of viewing and disparity on perceived depth. *Proceedings of the SPIE*, *1457*, 259-266. Bellingham, WA: SPIE.

Havaldar, P., Lee, M., & Medioni, G. (1997). Synthesizing novel views from unregistered 2D images. Computer graphics forum, 16 (1), 65-73.Heid 1996

Holloway, R. L. (1997). Registration error analysis for augmented reality. Presence, 6 (4), 413-432.

Horry, Y., Anjyo, K., & Arai, K. (1997). Tour into the picture: using a spidery mesh interface to make animation from a single image. Proceedings SIGGRAPH'97, 225-232.

Horvitz, E., & Lengyel, J. (1997). Perception, attention, and resources: a decision-theoretic approach to graphics rendering. Proceedings of the thirteenth conference on uncertainty in artificial intelligence (UAI '97, Providence, RI, August 1-3), 238-249. Available FTP: ftp.research.microsoft.com/pub/ejh/dtgraph.ps.

Jacoby, R. H., Adelstein, B. D., & Ellis, S. R. (1996). Improved temporal response in virtual environments through system hardware and software reorganization. Proceedings of the SPIE (28 January-2 February 1996, San Jose, CA), 2653, 271-284. Partly available Internet: http://duchamp.arc.nasa.gov/research/latency.html.

Kalman, R. E., & Bucy, R. S. (1961). New results in linear filtering and prediction theory. Trans. ASME, J. Basic Eng., Series 83D, 95-108.

Kano, H. (1995?). The CMU Video-Rate stereo machine. Carnegie-Mellon University. Available Internet: www.cs.cmu.edu/afs/cs/project/stereo-machine/www/z-key.html.

Keran, C. M., Smith, T. J., Koehler, E. J., & Mathison, P. K. (1994). Behavioral control characteristics of performance under feedback delay. Proceedings of the human factors and ergonomics society 38th annual meeting, p. 1140-1144.

Lengyel, J., & Snyder, J. (1997). Rendering with coherent layers. Proceedings of theSIGGRAPH'97, 233-242.

Levoy, M., & Hanrahan, P. (1996). Light field rendering. Computer graphics proceedings (SIGGRAPH'96), 31-42.

Lewis, F. L. (1986). Optimal estimation. New York: John Wiley & Sons, Inc.

List, U. (1983). Nonlinear prediction of head movements for helmet-mounted displays. U.S. Air force Human Resources Laboratory, Technical paper AFHRL-TP-83-45, December.

McWhorter, S. W., Hodges, L. F., & Rodriguez, W. E. (1991). Comparison of 3-D display formats for CAD applications. SPIE 1457: Stereoscopic displays and applications II, 85-90.

Mann, S. (1995). Wearable computing: A first step toward [sic?] personal imaging. IEEE Computer, 30 (2), February, 25-32.

Mann, Y., & Cohen-Or, D. (1997). Selective pixel transmission for navigating in remote virtual environments. Eurographics'97, 16 (3), C201-6.

McMillan, L., & Bishop, G. (1995). Head-tracked stereoscopic display using image warping. Proceedings of the SPIE (San Jose, CA, 5-10 February), 2409, 21-30.

Molnar, S., Eyles, J., & Poulton, J. (1992). PixelFlow: High-speed rendering using image composition. Proceedings of the SIGGRAPH'92, 231-240.

Montrym, J. S., Baum, D. R., Dignam, D. L., & Migdal, C. J. (1997). InfiniteReality: A real-time graphics system. Proceedings of the SIGGRAPH'97, 293-301.

Olano, M., Cohen, J., Mine, M., & Bishop, G. (1995). Combatting rendering latency. Proceedings of the 1995 symposium on interactive 3D graphics (Monterey, CA, April 9-12), 19-24 and 204.

Ono, H., Rogers, B. J., Ohmo, M., & Ono, M. E. (1988). Dynamic occlusion and motion parallax and absolute-distance information. Perception, 17, 255-266.

Padmos, P., & Milders, M. V. (1992). Quality criteria for simulator images: A literature review. *Human Factors, 34* (6), 727-748.

Pasman, W. (1997). *Enhancing x-ray baggage inspection by interactive viewpoint selection.* Doctoral dissertation, Faculty of Industrial Design Engineering, Delft University of Technology, The Netherlands. ISBN 90-9010959-5.

Quantum3D (1997). Obsidian 100SB. Available Internet: www.quantum3d.com.

Regan, M., & Pose, R. (1994). Priority rendering with a virtual address recalculation pipeline. Proceedings of the SIGGRAPH'94 (Orlando, FL, 24-29 July). In Computer Graphics, Annual conference series, 155-162.

Reinhart, W. F., Beaton, R. J., & Snyder, H. L. (1990). Comparison of depth cues for relative depth judgments. Proceedings of the SPIE 1256: Stereoscopic displays and applications, 12-21.

Sedgwick, H. A. (1986). Space perception. In Boff, K. R., Kaufman, L., & Thomas, J. P. (Eds.), *Handbook of perception and human performance* (Chapter 21). New York: John Wiley & Sons.

Seitz, S. M., & Dyer, C. R. (1996). View morphing. Proceedings of the SIGGRAPH'96, 21-31.

Serdick, R. T., Davis, E. T., King, R. A., & Hodges, L. F. (1997). The perception of distance in simulated visual displays: A comparison of the effectiveness and accuracy of multiple depth cues across viewing distances. Presence, 6 (5), 513-31.

Spain, E. H., & Holzhausen, K. P. (1991). Stereoscopic versus orthogonal view displays for performance of a remote manipulation task. *Proceedings of the SPIE, 1457*, 103-110. Bellingham, WA: SPIE.

Szeliski, R., & Shum, H (1997). Creating full view panoramic image mosaics and environment maps. Proceedings of the SIGGRAPH'97, 251-258.

The Virtual Reality Source (1997) [Internet page]. Arvada, CO 80002. Available Internet: http://www.thevrsource.com/tracker/flock.htm.

Torborg, J., & Kajiya, J. T. (1996). Talisman: Commodity realtime 3D graphics for the pc. Computer graphics proceedings (SIGGRAPH'96), 353-363. Available Internet: www.research.microsoft.com/SIGGRAPH96/96/Talisman.

Uenohara, M. (1997). Magic eye project. Carnegie Millan University, Department of Computer Science. Available Internet: www.cs.cmu.edu/afs/cs/user/mue/www/magiceye.html.

Vallino, J. R. (1997). Augmented reality page (version of 30 October). University of Rochester, NY, Department of Computer Science. Available internet: http://www.cs.rochester.edu:80/u/vallino/research/AR.

Wang, J., Azuma, R., Bishop, G., Chi, V., Eyles, J., & Fuchs, H. (1990). Tracking a Head-Mounted Display in a Room-Sized Environment with Head-Mounted Cameras. SPIE Proceedings Vol. 1290 Helmet-Mounted Displays II (Orlando, FL, 19-20 April 1990), 47-57.

Ward, M., Azuma, R., Bennett, R., Gottschalk, S., & Fuchs , H. (1992). A Demonstrated Optical Tracker With Scalable Work Area for Head-Mounted Display Systems. Proceedings of 1992 Symposium on Interactive 3D Graphics (Cambridge, MA, 29 March - 1 April 1992), 43-52. Abstract Available Internet: www.cs.unc.edu/~azuma/cambridge.abstract.html.

Welch, G., & Bishop, G. (1997). SCAAT: Incremental trackingwith incomplete information. Proceedings of the SIGGRAPH'97, 333-344. See also Internet: www.cs.unc.edu/~welch/scaat.html.

Wickens, C. D. (1990). Three-dimensional stereoscopic display implementation: Guidelines derived from human visual capabilities. *Proceedings of the SPIE, 1256*, 2-11. Bellingham, WA: SPIE.

Wloka, M. M. (1995). Lag in multiprocessor virtual reality. *Presence, 4* (1), 50-63.