N95- 15977

VIRTUAL REALITY AND TELEROBOTICS APPLICATIONS OF AN ADDRESS RECALCULATION PIPELINE.

Matthew Regan and Ronald Pose.

Department of Computer Science, Monash University,
Clayton, Victoria 3168, Australia.
Email: {regan,rdp}@cs.monash.edu.au

ABSTRACT.

The technology developed at Monash University and described in this paper was designed to reduce latency to user interactions in immersive virtual reality environments. It is also ideally suited to telerobotic applications such as interaction with remote robotic manipulators in space or in deep sea operations. In such circumstances the significant latency in observed response to user stimulus which is due to communication delays, and the disturbing jerkiness due to low and unpredictable frame rates on compressed video user feedback or computationally limited virtual worlds, can be masked by our techniques. The user is provided with highly responsive visual feedback independent of communication or computational delays in providing physical video feedback or in rendering virtual world images. Virtual and physical environments can be combined seamlessly using these techniques.

INTRODUCTION.

The combination of Delayed Viewport Mapping [1] as implemented using an Address Recalculation Pipeline, image composition [2], and Prioritized Rendering [3] provides not only an order of magnitude reduction in image rendering required for interaction with a given virtual world, but a useful tool for all head mounted applications. A further important benefit is the graceful handling of inadequate computational capacity without sacrificing image resolution or latency to interaction.

An Address Recalculation Pipeline (ARP)[1] is a hardware implemented algorithm which performs delayed viewport orientation mapping. Using an ARP it is possible to orientate a computer generated virtual world with a user's head orientation after the scene has been rendered rather than before, as is the case with conventional virtual reality systems. This drastically reduces the computational component of the latency perceived by the user. Latency to user head rotations is essentially removed and latency to user translations may be significantly reduced with the use of image composition and priority rendering.

With image composition a scene is divided into several sections, each being allocated to a different rendering engine. Thus there are several rendering engines drawing different parts of a scene into different display memories in parallel. When displaying the scene, the images in all of the display memories are composed. All of the pixels spread across the display memories which correspond to a screen location in the head mounted display or other display device, are fetched simultaneously and the pixel with the smallest Z-value is displayed. Conventional systems can achieve almost linear speedup with multiple rendering engines [2].

With the viewport independence provided by the ARP in a head mounted display environment prioritized rendering [3] can be employed. With this scheme one can update the different display memories at different rates making it possible to render only those parts of the scene which change or which are most important to update quickly rather than the entire scene. Note that this is independent of interactive latency.

Experiments have shown that the speed up achieved with prioritized rendering can be significant. In a sample virtual world the number of objects redrawn at any update was reduced by on average 90% [3]. Thus for M rendering engines we achieve a speedup of 10 M.

The low latency achieved by these methods based around an ARP may be applied to a augmented reality and telerobotics applications. In a telerobotic environment the latency to user head rotations tends to be quite high. The mechanical delays involved in making a robotic head follow the motion of a user's head are a significant component however the additional two-way communications delay is also important.

Using an ARP it is possible to correct for the difference in orientation between the most up to date head tracking information and the orientation of the robotic head when the image was captured. As a result the user may see an old image which has been remapped to compensate for its invalid orientation. The user sees all images with the correct orientation at the update rate of the display device, typically 60Hz. The ARP does not need to be updated at the maximum rate or even a predictable rate. Image compression which generally leads to unpredictable frame rates may be used without annoying side-effects.

THE ADDRESS RECALCULATION PIPELINE.

An address recalculation pipeline is a hardware implemented algorithm which performs viewport orientation mapping after rendering. Rather than using a simple conventional counter for display memory access the addressing mechanism becomes quite complex and provides a correction for wide angle viewing lenses and user head orientation as pixels are fetched from display memory.

The user head orientation doesn't need to be known accurately until the first pixel of a frame is to be displayed on the output device. As a result the latency to user head rotations caused by computational delays is in the order of two microseconds. This latency is independent of scene complexity and renderer overload.

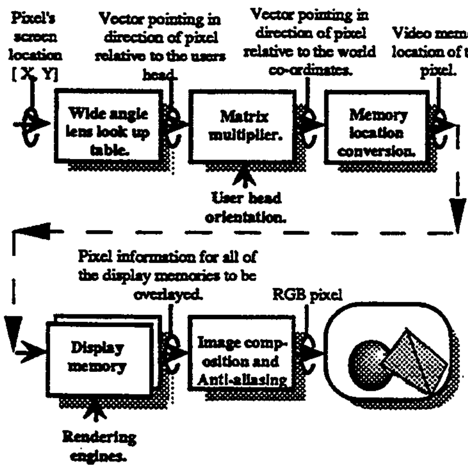
Pixel's screen location [X, Y]
Vector pointing in direction of pixel relative to the users head.
Vector pointing in direction of pixel relative to the world co-ordinates.
Video memory location of the pixel.
Wide angle lens look up table.
Matrix multiplier.
Memory location conversion.
User head orientation.
Pixel information for all of the display memories to be overlayed.
RGB pixel
Display memory
Image composition and Anti-aliasing
Rendering engines.

Figure 1. The address recalculation pipeline.

The pipeline is depicted in Figure 1. The first input to the pipeline is the X-Y screen location of the current pixel to be displayed. This screen location is provided by a conventional graphics controller at a normal pixel display rate. A look up table called a Wide Angle Viewing Lens Lookup Table converts the screen X-Y location into a 3-D unit vector pointing in the direction at which the pixel is seen by the user, through the wide angle lenses, relative to the user's head. The look up table has one entry per display pixel where each entry consists of three 16-bit vector components. For a display device of resolution 640 by 480 pixels the lookup table will require a memory of 1024(cols)*512(rows)*6(bytes per entry) = 3 Mbytes. Many head mounted displays use wide angle viewing lenses which preserve a standard viewport mapping, however if a special mapping is required for higher fields of view[6][7], the lookup table may be loaded with a new lens mapping, compensating for the lens mapping without run-time penalty.

The 48-bit output of the wide angle viewing lens feeds into a matrix multiplier which forms the next stage of the pipeline. The multiplier multiplies the pixel direction vector with a 3 by 3 matrix containing user head orientation information. The resulting output vector points in the direction at which the pixel is seen by the user, through the wide angle viewing lenses, relative to the world coordinate system. The pixel direction vector is fed into the matrix multiplier at pixel display rates while the head orientation matrix is updated at the start of each display frame (i.e.. after each vertical sync signal). The matrix multiplier is also implemented with 16-bit fixed point arithmetic and is built with nine commercially available 16-bit by 16-bit, 40ns multipliers and six 16-bit, 40ns adders. The output vector from the matrix multiplier is in the form of three 16-bit fixed point vector components [Vx Vy Vz].

The next pipeline stage, called the Vector Conversion stage, converts the 3D unit vector into a display memory location. The chosen display memory topology for this architecture is the surface of a cube. A spherical topology is also possible [5]. Figure 2 depicts an the face organization on the surface of a cube. The conversion process involves computing the point at which the ray intersects with the surface of a cube. When the cube is aligned to the axes of the coordinate system such that each face of the cube has one of its X, Y or Z coordinates fixed at +/- 1.0, the intersection may be computed with a set of parallel divisions with range checks on the outputs of the divisions. For example if the result of the divisions Vx/Vy and Vz/Vy are both within the range (-1.0, 1.0) the ray must intersect with only two of the six faces. The sign of Vy is then used to determine the face of intersection. The point of intersection on the face is then (Vx/Vy, Vz/Vy). The divisions must occur at pixel display rates, so the divisions are performed by a reciprocal lookup followed by a normal multiply using another set of 40ns multipliers. The reciprocal lookup has extra output bits which are used to compensate for classification with fixed precision arithmetic. A programmable logic device is used to accumulate data from the appropriate data paths to multiplex the divider outputs to form the display memory address.

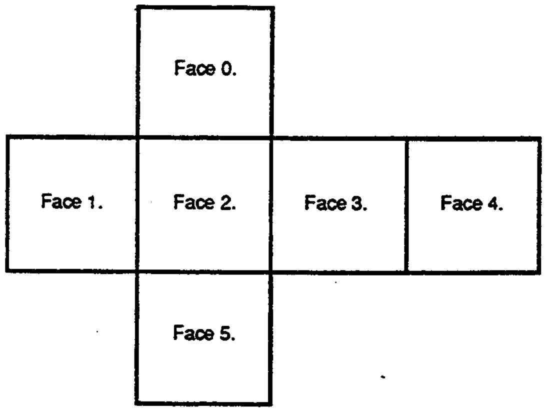

Figure 2. Display Memory Organization.

The address recalculation pipeline performs a remapping of the image in display memory to form an output image in real time based on the wide angle viewing lenses and the user head orientation. This remapping is performed one pixel at a time by the hardware in the address recalculation pipeline. The remapping occurs by sampling the image contained within the display memory and as with any form of discrete sampling, aliasing occurs. Even if the image in the display memory is anti-aliased and rendered with a high quality rendering technique, the hardware sampling occurring will cause aliasing in the final image. The aliases introduced by the address recalculation pipeline cannot be corrected with software in the rendering process. Any pipeline anti-aliasing must occur in hardware.

After simulating the possible artifacts caused by no hardware antialiasing strategy and considering the overall cost of an address recalculation pipeline system, hardware anti-aliasing is considered necessary. The anti-aliasing strategy chosen for this architecture is a linear interpolation filter using redundant addressing bits from the intersection computation. A linear interpolation filter provides an adequate trade-off between system expense and filter quality[8]. In order to perform linear interpolation the four pixels surrounding the point of intersection must be fetched simultaneously. The interleaving mechanism for fetching the four adjacent pixels and the method of interpolation are discussed in detail in [4].

VIRTUAL REALITY.

The ARP was designed specifically to compensate for many of the problems associated with HMD graphics systems. Image composition opens up a gateway into priority rendering which leads to significant gains in the effective rendering performance of a virtual reality graphics system.

Image overlaying or image composition [2] is a technique often used to increase the apparent display memory bandwidth as seen from the renderer. Rather than having one display memory (or two for double buffering) the graphics system has multiple display memories. Different sections of the visible scene may drawn into separate display memories then overlaid to form a final scene. In many implementations each display memory has a private rendering engine. The concept of image composition is depicted in figure 3.

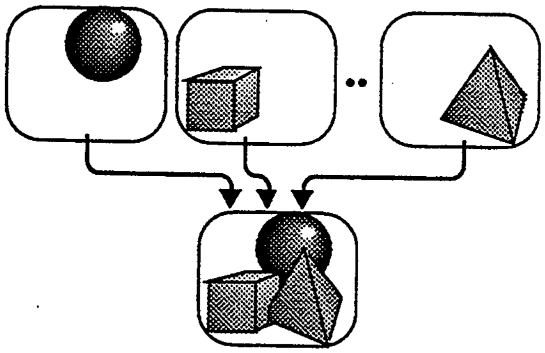

Figure 3. Image Composition.

Image composition allows the possibility of rendering different objects (down to a polygonal level) to different display memories. A side-effect of image composition is that each display memory may have its own unique update period. Using an ARP it is possible to make effective use of this side effect of image composition to achieve in certain cases much better performance increases from the available rendering hardware when compared to conventional image composition systems. This performance improvement eludes the conventional systems because the images in the display memories of a graphics system with an ARP do not necessarily become invalid when the user's head orientation changes, thus the length of time an image in display memory is valid only loosely depends on the orientation (for a stereo view). For example a non interactive background may never require re-rendering and may thus be pre-rendered with great detail using a high quality rendering technique and a complex model.

Using an ARP it is possible to render a scene which is largely independent of the user's head orientation. When image composition is combined with the address recalculation pipeline it is possible to render different parts of a scene at different rates this new paradigm is called priority rendering. Priority rendering is demand driven rendering. An object is not redrawn until its image within the display memory has changed by a predetermined threshold. In a conventional system this strategy would not be effective as almost any head rotations would cause considerable changes to the image in display memory and the system would have to re-render everything.

The threshold for determining when an object has changed by more than a tolerable amount is determined by the designer of the virtual world and may typically be based on several factors. Usually this threshold is in the form of an angle (θ_t) which defines the minimum feature size of the world. This value may vary from less than the minimum feature size the human eye can detect, to the size of one or more display pixels. Priority rendering attempts to keep the image in display memory accurate to within θ_t at the highest possible update rate.

In order to compute the period for which a given object is valid we compute the time it takes for the object to translate by θ_t, to grow or shrink by θ_t or to change by θ_t due to animation. The objects validity period may be computed from the size of the object, the distance to the object and the user's speed relative to that object. An additional factor is added by the designer of the virtual world which describes how much the object is animating. Once the period for which an object's image is valid has been determined the object may be assigned to a display memory with an appropriate update rate. A more detailed explanation of the computation of validity and assignment to display memories is given in [3]

The rendering hardware may have more display memories available than the virtual world requires for high efficiency. In this event, multiple renderers and display memories may be assigned to the one update rate thus devoting more hardware resources to a particular update rate, helping to balance the load.

Priority rendering may be used to reduce the overall rendering load on the rendering subsystem. The rendering load is based on several features of the scene, where the actual number of polygons is just one of the factors. One of our virtual world applications is a walk through of a forest. This simulation was performed in order to determine the rendering load on various display memories with various update rates.

In the experimental virtual environment the combination of the ARP, image composition and priority rendering cut the total number of objects requiring re-rendering by 90% when compared with the number of objects requiring re-rendering in an equivalent system without the ARP. That is, the system with the pipeline only had to redraw 10 objects for every 100 objects the system without the pipeline had to redraw for a similar illusion.

A stereo view of a virtual world is highly desirable within a head mounted graphics system. With an ARP the display memories are not actually centered around the point of rotation of the user's head, rather they are centered around the user's eyes. This means when the user's head rotates while the user is stationary a small amount of translation occurs. This implies the need to re-render some objects which are affected by the translation caused by the head rotation. Experiments have shown that head rotations smaller than 45 degrees require few objects to be updated due to the small translation of the eyes[3].

AUGMENTED REALITY.

An ARP graphics system with priority rendering may be used for augmented reality applications in the same way it is used for virtual reality applications, however the use of an ARP alone has significant advantages over a conventional graphics system when applied to augmented reality environment graphics systems.

The difference between virtual reality and augmented reality is in their treatment of the real world. Virtual reality environments immerse the user inside a virtual world that completely replaces the real world outside. In contrast augmented reality uses see-through HMDs that let the user see the real world and the virtual world at the same time. See-through HMDs augment the user's view of the real world by overlaying or compositing three-dimensional virtual objects with their real world counterparts. Ideally, it would seem to the user that the virtual and real objects co-exist.

Researcher in the field of augmented reality recognize that to use the technology in practice the 'registration problem' must be overcome[9]. The real and virtual objects must be aligned with respect to one and other, or the illusion that the two co-exist will be compromised.

The main sources of registration errors are,
-Distortions in the HMD optics.
-End-to-end system latency.
-Mechanical misalignment in the HMD.
-Errors in the head tracking system.
-Incorrect viewing parameters (field of view, tracker-to-eye position and orientation, interpupillary distance)
Of these factors, only the first two factors may by improved by modifying the image generation process alone.

Distortions in the HMD optics in an augmented reality environment become particularly noticeable when the distortion of the image from the image generator does not match the view of the real world. Conventional real-time image generation systems tend not to provide a facility to correct for the distortions introduced by the optics in a HMD. If there is some form of distortion correction, it is usually prohibitively expensive.

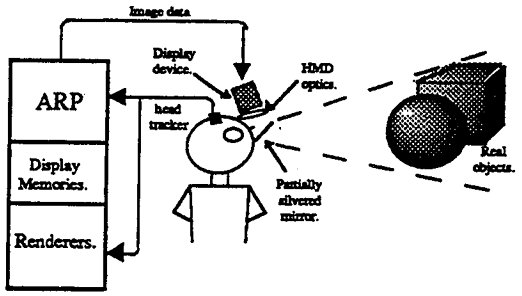

Figure 4. An augmented Reality environment.

An ARP provides a mechanism for correcting optical distortions introduced by optics within the HMD system. This mechanism derives from the versatility of the Wide Angle Viewing Lens Look Up Table (WAVELUT). The WAVELUT is a large lookup table which contains a direction vector for each pixel on the output display device. Provided the nature of the optical distortion of the HMD optics is known, the direction at which each pixel is seen by the user relative to the HMD may be computed. For each pixel in the displayed output an associated unit direction vector is computed and downloaded into the lookup table. The computation of this distortion is a one-off expense and allows for real-time correction of optical distortions without penalizing rendering performance. When new optics are installed in the HMD, or a new HMD is to be used, the optics need to be recomputed once. Such a setup is depicted in Figure 4.

A major feature of the ARP is that the update rate for user head rotations is bound to the update rate of the display device usually 60+ Hz, instead of the rendering frame rate. Also, with an ARP, the latency does not include the rendering time and doesn't include double buffer swap delays. The orientation of the view the user sees does not need to be known until the first pixel is to be sent to the display device. This means the images the user sees use the most up to date head tracking information. The nature of the latency to head rotations is depicted in Figure 5.

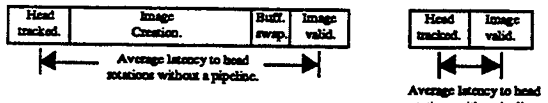
Head tracked.
Image Creation.
Buff. swap.
Image valid.
Average latency to head rotations without a pipeline.
Head tracked.
Image valid.
Average latency to head rotations with a pipeline.

Figure 5. Latency to head rotations.

As a result end-to-end latency performance is improved when compared with conventional augment reality systems by by-passing the image generation component of the latency period for head rotations. The latency induced by the ARP is effectively less than two micro seconds and therefore may be neglected.

The magnitude of the end-to-end latency in an augmented reality environment is more critical than any other HMD application, such as VR or telerobotics[10]. This is because the user's visual system has real world objects which have zero latency to act as references against the virtual objects. An ARP is effectively capable of removing the latency for head rotations greatly improving the registration between the real and virtual objects.

The hardware in the ARP allows compensation for head orientation changes only, however priority rendering may be used to improve the performance of the system when user translations occur similar to the case of the conventional virtual reality environments.

TELEROBOTICS.

Telerobotics technology is a powerful way of allowing machines under human control to operate in environments that are hostile to humans. The goal of the technology is to convince a user that he or she is in the hostile environment to such a degree that the human user may perform complex operations through the robot, that a robot could not perform autonomously. Applications vary from robots in deep space building space stations to deep sea applications were robots repair and maintain underwater pipelines.

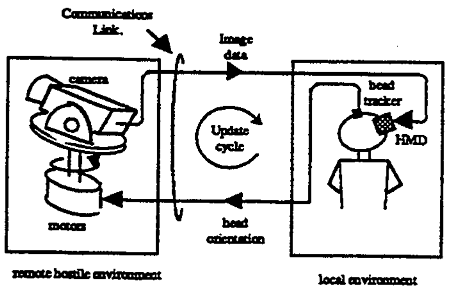

Figure 6. A conventional telerobotic setup.

Conventional telerobotic techniques usually feed the displays in an HMD from a set of cameras on a remote robotic head. The orientation of the robotic head is controlled with the information gained from the user head tracking equipment following the motion of the user's head. Ideally the motion of the robotic head would match the motion of the user's head. Figure 6 depicts a typical telerobotic setup with a HMD. End-to-end latency tends be exceptionally high in such a scenario. Once the user's head position has been tracked the orientation is sent via a potentially long communication path to motors controlling the orientation of the robotic head. These physical motors respond, move the robotic head to the desired position and capture an image. Next the image is sent back via the communications path to the HMD where it is displayed to the user. It becomes quite clear that even if very fast motors are used with short, high speed communications paths, the latency to orientation changes by the user's head will be very high.

Latency in a telerobotic environment is caused by a mismatch in the orientation of the users head and the orientation of the view the user sees in the HMD. The orientation of this view is the same as the orientation of the robotic head at some previous time. With an ARP it is possible to correct for the difference between the orientation of the robotic head when the image was captured and the current orientation of the user's head. Instead of feeding the image from the robotic head to the displays in the HMD, it is sent to the display memory of an ARP graphics system. At the start of each user update cycle, the users head orientation matrix is multiplied by an orientation matrix from the robotic head (which is obtained from sensors on the motors which move the robotic head) and represent the orientation of the real-world relative to the robotic head at the time when the image in the display memory was captured. The result is a matrix which converts the users head orientation to the robotic head coordinate system. This matrix is then fed into the ARP at the start of each user update. Such a setup is shown in Figure 7. As the display memory is divided into six faces, it is necessary to have multiple cameras on the robotic head to capture the entire image. The actual number of cameras required depends on the latency of the robotic update cycle however it is assumed that the number of cameras required is between four and six. This paper will not go into the physical details of the camera arrangement.

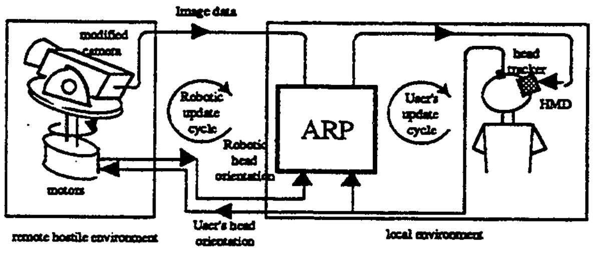

Figure 7. A telerobotic setup with an ARP.

The images in the systems display memory are valid for more than one update of the HMD, and as such it is possible to use the same image for a quite some period of time, this is similar to priority rendering in virtual reality where the user sees images that have been valid for some period of time. So while the orientation of the image the user sees is being updated at the frame rate of the output device, i.e. 60+Hz (user update rate), it is possible that the image stored in display memory is being updated at a lower rate, i.e. 5-30Hz (robotic update rate). No matter how long the communication delay to and from the telerobotic environment may be, the user always sees images with the correct orientation.

The update rate for the images coming from the robotic head need not match the update rate the user apparently sees. More importantly, the rate at which the images come from the robotic head need not even be predictable. As such the images from the robotic head may be compressed for transmission. The ARP will keep using the image last received until a new image has arrived. In a conventional system, such unpredictability of frame rate would be very noticeable especially when the user is looking at the most complex objects (hardest to compress). The use of image compression is extremely desirable when the communications path is long and expensive.

While the ARP does reduce the rotational latency, there will be latency to translations. To perform translations the robotic head actually needs to move, hence the latency will be bound to the robotic update cycle. However, latency to translations are common in everyday life, for example there is a delay between pressing on an accelerator in a car and motion of that car, such delays are easily tolerated by humans. Latency to head rotations are not seen in everyday life and may be very disorientating.

Applications which require a stereo view of the remote environment require 2 camera setups. Some latency to stereoscopy will be noticed as the user's head rotates, however latency to stereoscopy is easily tolerated by humans.

CONCLUSION.

In this paper we have described the how an ARP achieves low latency to head rotations in all HMD environment, and how it leads to an order of magnitude reduction in rendering costs, how it improves registration within an augmented reality system and how it may be used to hide often lengthy mechanical and communications delays in a telerobotic environment.

The low cost and high performance of an ARP makes it an ideal interface to a head mounted display. Whether the application is virtual reality, augmented reality or telerobotics the ARP has clear advantages over conventional systems.

ACKNOWLEDGMENTS.

Matthew Regan acknowledges the support of a priority Australian Postgraduate Research Award. This work has been supported by an Australian Research Council small research grant.

REFERENCES.

1. Regan, M. and Pose, R., "An Interactive Graphics Display Architecture". In proceedings of IEEE Virtual Reality Annual International Symposium, VRAIS 93, (Sept 18-23, 1993) Seattle WA. pp. 293-299

2. Molnar. S, "Image Composition Architectures for Real-Time Image Generation" Ph.D. Dissertation, University of North Carolina, 1991.

3. Regan, M. and Pose, R., "Priority Rendering with Virtual Reality Address Recalculation Pipeline". In proceedings of Siggraph 94. pp.155-162.

4. Regan, M. and Pose, R., "Display Memory Access Issues and Anti-Aliasing with a Virtual Reality Graphics Controller". Proceedings of the 9th Eurographics workshop on graphics hardware, (Sept 12-14, 1994), Oslo, Norway.

5. Regan, M. and Pose, R., "A Low Latency Virtual Reality Display System". Monash University, Department of Computer Science, Technical Report TR166, September 1992.

6. Robinett, W., and Roland, J., "A Computational Model for the Stereoscopic Optics for Head Mounted Display". In Presence 1, (winter 1992), pp. 45-62.

7. Deering, M., "High Resolution Virtual Reality", In proceedings of Siggraph 92, pp. 195-202.

8. Wolberg, G. "Digital Image Warping" IEEE Computer Society Press, 1990.

9 Azuma, R., and Bishop, G., "Improving Static and Dynamic Registration in an Optical See-through HMD", In proceedings of Siggraph 94, pp. 197-204.

10. Azuma, R., "Tracking Requirements for Augmented Reality.", CACM 36, 7 (July 1993), pp50-51.