Low latency rendering for mobile augmented reality

W. Pasman, A. van der Schaaf, R.L. Lagendijk, F.W. Jansen
Ubicom-project, Faculty of Information Technology and Systems, Delft University of Technology, The Netherlands

Keywords Augmented Reality, latency, alignment, information-on-the-spot

Abstract

Mobile augmented reality requires accurately alignment of virtual information with objects visible in the real world. We describe a system for mobile communications to be developed to meet these strict alignment criteria using a combination of computer vision and inertial tracking and image-based rendering techniques.

1 Introduction

Mobile augmented reality [1,2] is a relatively new and intriguing concept. The ability of augmented reality [3] to present information superimposed on our view on the world opens up many interesting opportunities for graphical interaction with our direct environment. Combining this with mobility further increases the potential usage of this technology for direct daily use.

However, the technical problems with mobile augmented reality are just as great. As with other head-mounted display systems, augmented-reality displays also require an extremely high update rate. Simple head movements may in short time give rise to significant changes in viewing position and viewing direction (figure 1).

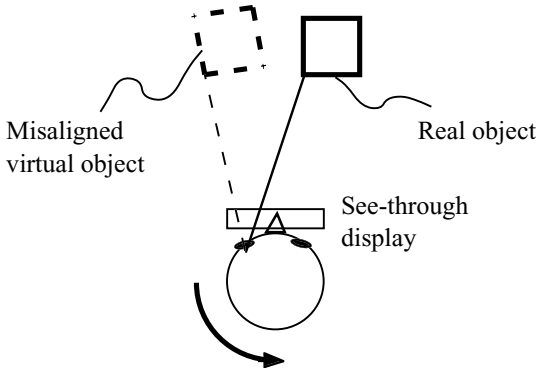

Figure 1. A virtual object is displayed in overlay with a real object. When the user rotates his head to the left, the real object immediately moves for the user to the right. The virtual object, however, has latency, and therefore is displayed for some time in the same direction it was before, and only after some time is re-rendered in alignment with the real object.

The virtual information associated with objects in the scene and displayed within the viewing window will then have to be updated to maintain the proper alignment with the objects in the real world. The viewpoint changes will therefore have to be tracked and fed back to the display system, in order to re-render the virtual information in time at the correct position.

Padmos and Milders [4] indicate that for immersive reality (where the observer can not see the normal world), the update times (lag) should be below 40 ms. For augmented reality the constraints will be much stricter. They suggest that the displacement of objects between two frames should not exceed 15 arcmin (0.25°), which would require a maximal lag of 5 ms even when the observer rotates his head with a moderate speed of 50°/s. Several other authors use a similar approach [5-9] and come to similar maximal lag times. Actually, during typical head motions speeds of up to 370°/s may occur [10], but it is not likely that observers rotating their head that fast will notice slight object displacements. Many authors suggest that 10 ms will be acceptable for AR [5,11,12]. Summarizing, we may say that the alignment criteria both for accurate positioning and for time lag are extremely high.

In this paper we describe how these strict requirements could be met with a combination of several levels of position and orientation tracking with different relative and absolute accuracies, and several levels of rendering to reduce the complex 3D data to simple image layers that can be rendered just-in-time. In section 2 we first describe the context of our research, the Ubicom project, a multi-disciplinary project carried out at Delft University of Technology, which aims at the development of a system for Ubiquitous Communication. In section 3 and 4 we focus on the problem of image stabilisation and discuss latency issues related to position tracking and display. We summarize our system set-up in section 5 and conclude with describing the current state in section 6.

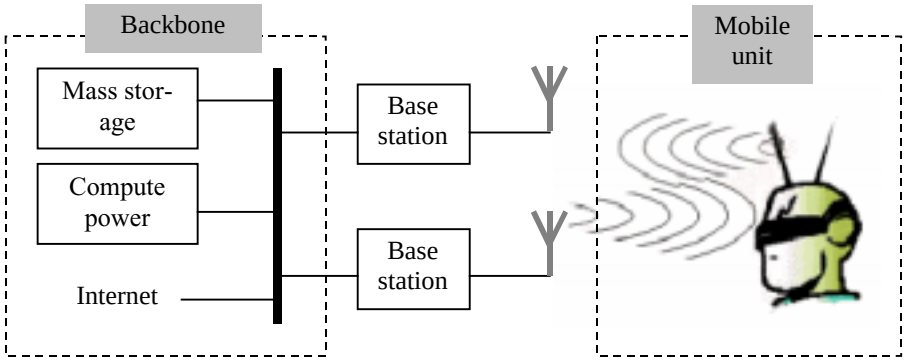

Figure 2: Ubicom system set-up. The mobile unit contains display, camera, and tracking devices, and is connected through a mobile link to one of several base stations. Memory and processing resources are limited in the mobile unit in order to reduce power consumption and extend battery life. Instead, the mobile connection is used to access resources like mass storage and compute power at the backbone.

2 Ubicom system

The Ubicom System [13] is an infrastructure for mobile multi-media communication. The system consists of a backbone compute server, several base stations, and a possible large number of mobile units (figure 2).

The base stations maintain a wireless (radio or infrared) link to the mobile units. The radio transmission is scheduled in the 17 GHz range and will account for approximately 10 Mbit/s of data bandwidth per user, enough to transmit compressed video with high quality.

The cell size (distance between the base stations) is in the order of 100 meter: typically the distance between lampposts to which the base stations may be attached.

The mobile unit consists of a receiver unit and a head-set. The head-set contains a light-weight head-mounted display that offers the user a mix of real and virtual in-

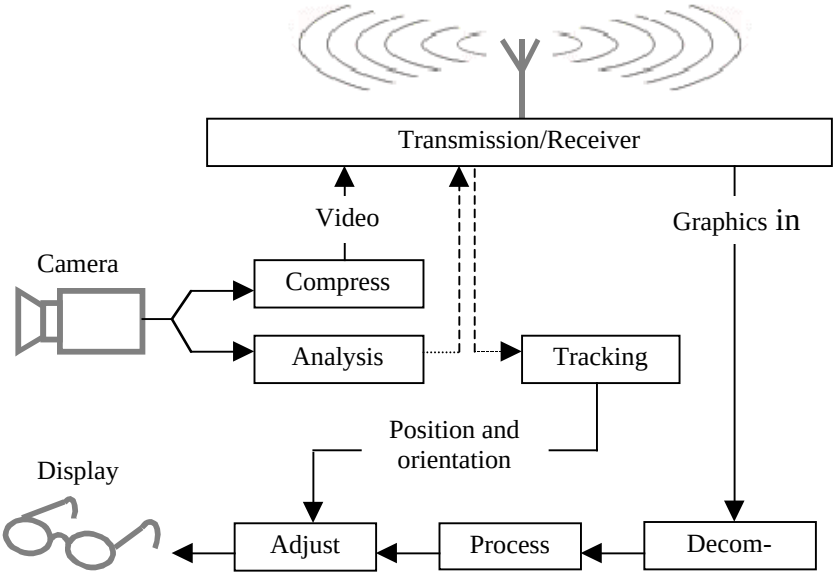

Figure 3: Diagram of the mobile unit. The camera at the mobile unit supports two main functions. First, the camera produces video, which is compressed and sent to the backbone for recording or distribution to other users. Second, the camera images are analysed to find landmarks that are used for position tracking. The actual matching of landmarks is computationally expensive and is done at the backbone. The backbone also supplies the mobile unit with the AR graphics, which must be decompressed and processed before they can be displayed in overlay with the real world. Fast inertial tracking devices in the mobile unit measure the head-motion of the user and track the latest position and orientation of the head-set. This information is used for last-minute adjustments of the displayed graphics, such that these remain in register with the real world.

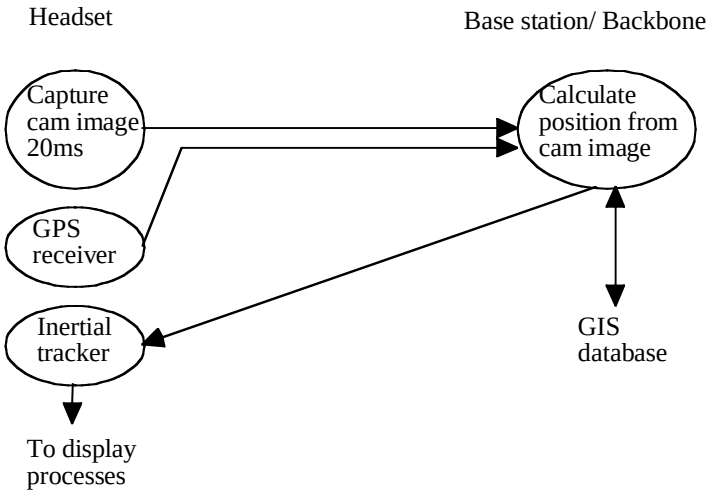

Figure 4. Circles represent processes, boxes local memory and arrows the data flow. Position changes are directly determined with the inertial tracker. To compensate for drift, more accurate positions are calculated regularly in the backbone, based on GPS data and camera images.

formation. This may be realised either superimposing the virtual information on the real world or by replacing parts of the real world with virtual information. In the latter case we need partially visual blocking of the view on the outside world. In addition to the display facilities, the headset will also have a light-weight video camera that is used for position tracking and to record video data. In order to keep the power consumption low, the head-set and receiver unit will only have limited processing and memory capabilities. Figure 3 shows the diagram of the head-set and receiver unit.

3 Tracking

Central to the function of the headset is the exact alignment of virtual information with the objects in the real world that the user is seeing. This requires that the exact viewing position and viewing direction of the user are known. Position as well as orientation tracking are therefore needed. Orientation tracking is much more critical than position tracking as a small rotation of the head will have a larger visual impact than a small movement to the left or right.

Position tracking is done in three steps (Figure 4). A first position estimation is done using GPS or similar position detecting techniques. A possibility is to calculate the position relative to the base stations. A second level of position tracking is using object and scene recognition. Given a 3D description of the environment (e.g. a 3D GIS or CAD-model) and an initial position estimate, an accurate position may be calculated iteratively. However, the model data will only be available at the backbone and most of the calculations to derive the viewing position will have to be performed at the backbone as well. Part of this computation could be offloaded to the active base station. The latency introduced by first sending the video captured scene information from the mobile unit to the backbone, then the processing at the backbone or base station and the transmission of the obtained viewing parameters, will be too large for the update of the visual display. Therefore to be able to anticipate on small position changes immediately, the direction and acceleration of the movement will be sensed with an inertial tracker and directly fed back to the display system. In the same way, the orientation tracking will be based on object recognition and direct feedback from the inertial tracker.

4 Low-latency rendering

Given an accurate viewing position, a new virtual image will have to be generated. Also here the choice is whether to calculate each new image at the backbone with a powerful render engine and to transmit the image to the mobile unit over the wireless link, or to render the image directly by the mobile unit, avoiding the latency of the wireless link. Even for the second option, direct rendering at the headset with standard rendering hardware, there will be a latency in the order of 50-100 ms, which is unacceptable.

To compensate for small changes in perspective and viewing direction, we could apply image warping and viewport re-mapping techniques [14,15]. To further account for parallax changes, the virtual and real-world information could be segmented in layers, and the resulting image would be calculated by merging the warped image layers [16] (Figure 5).

In order to be able to generate these image layers within certain time constraints, we first segment the model data in model layers to reduce model complexity. The model simplification could be done at the backbone while the image layer rendering - taking into account the current

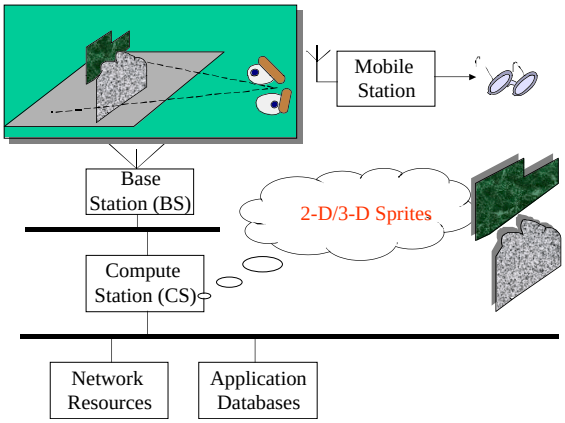

Figure 5. The virtual scene is decomposed in image layers. Given a new view point the layers are combined to form a new image.

view point - could be off-loaded to the active base station.

5 Conclusions

If we analyze the latency of the inertial tracking and corresponding image compensation, we come to the following conclusions (Figure 6).

In global, we have three paths to refresh the image in the head-set with increasing latency times and increasing accuracy: a path local to the headset, a path from head-set to base station and back, and a path from head-set via base station to the backbone and back. In the headset we minimise latency by using an inertial tracker (2 ms delay) and image warping and combining (8 ms). The image warping and combining is done just ahead of the display scanning, to avoid latency that might be caused by the refresh rate of the display (Figure 7).

In the base station, a simplified virtual world is being

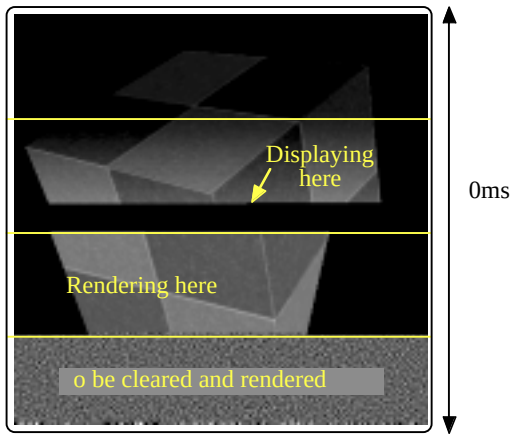

Figure 7. Display time for the whole image is 20 ms, excl. rendering. Dividing the image into partitions, and rendering just ahead of the partition being displayed, reduces the latency to 10 ms (5 ms rendering and 5 ms display).

rendered to images that can be used in the headset. Either the headset itself requests for these images or the base station anticipates the need for new images from recent movement data passing through the base station. These new images will have a lag of about 200 ms when arriving at the headset. In the backbone there are two processes. The first calculates the viewpoint of the observer given camera images from the headset and a GIS database. This process may be supported by GPS data acquired in the headset, and may take up to 500 ms including all transmissions back to the headset. The sec-

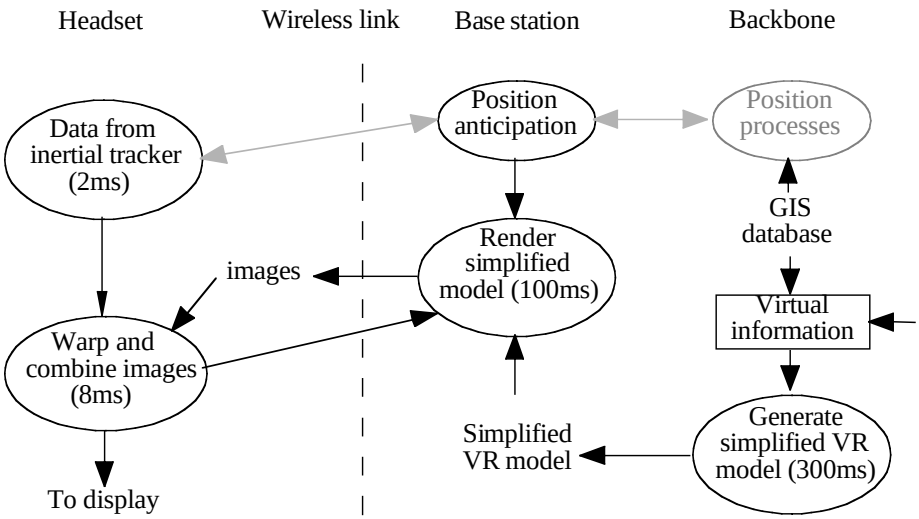

Figure 6. The rendering is distributed over headset, compute station and backbone, giving different latencies for these parts of the rendering process.

ond process is the generation of a new virtual world by generating 3D model layers. Images generated from the new simplified virtual world model rendered at the base station will arrive at the headset with a latency of about 1000 ms, one second.

6 Current state

The current state of the research is that these techniques will be implemented on a hardware prototype system built from off-the-shelf components. The system will have a limited portability, an experimental infrared mobile link, and it will use a standard look-through head-mounted display. The system will be tested within our research lab environment. The system will be ready in March '99 and the rendering software will be runnning a few months later.

References

[1] Feiner, S. (1995). KARMA. http://www.cs. columbia.edu/graphics/projects/karma.

[2] 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), www.cs.columbia.edu/ graphics/publications/ISWC97.ps.gz.

[3] Milgram, P. (1995). Augmented Reality. http: //vered.rose.utoronto.ca/people/anu_dir/papers/atc/ atcDND.html.

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

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

[6] Azuma, R. T. (1997a). 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. http://www.cs.unc.edu/~azuma/ ARpresence.pdf.

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

[8] 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.

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

[10] 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.

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

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

[13] Ubicom (1997), http://www.ubicom.tudelft.nl.

[14] 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.

[15] Mark, W. R., McMillan, L., Bishop, G. (1997). Post-rendering 3D warping. Proceedings of the 1997 Symposium on Interactive 3D Graphics (Providence, RI, April 27-30), 7-16. http: //www.cs.unc.edu/~billmark/i3dwww/i3dpaper-web.pdf.

[16] Lengyel, J., Snyder, J. (1997). Rendering with coherent layers. Proceedings of the SIGGRAPH'97, 233-242.