

# Dynamic Registration Correction in Augmented-Reality Systems

*Michael Bajura*  
Department of Computer Science  
UNC Chapel Hill  
Chapel Hill, NC 27599-3175  
bajura@cs.unc.edu

*Ulrich Neumann*  
Computer Science Department  
University of Southern California  
Los Angeles, CA 90089-0781  
uneumann@cs.usc.edu

## ABSTRACT

This paper addresses the problem of correcting visual registration errors in video-based augmented-reality systems. Accurate visual registration between real and computer-generated objects in combined images is critically important for conveying the perception that both types of object occupy the same 3-dimensional (3D) space. To date, augmented-reality systems have concentrated on simply improving 3D coordinate system registration in order to improve apparent (image) registration error. This paper introduces the idea of dynamically measuring registration error in combined images (2D error) and using that information to correct 3D coordinate system registration error which in turn improves registration in the combined images. Registration can be made exact in every combined image if a small video delay can be tolerated. Our experimental augmented-reality system achieves improved image registration, stability, and error tolerance from tracking system drift and jitter over current augmented-reality systems. No additional tracking hardware or other devices are needed on the user's head-mounted display. Computer-generated objects can be "nailed" to real-world reference points in every image the user sees with an easily-implemented algorithm. Dynamic error correction as demonstrated here will likely be a key component of future augmented-reality systems.

**KEYWORDS:** Augmented Reality, Virtual Reality, Registration.

## 1. INTRODUCTION

Augmented-reality (AR) systems allow users to interact with real and computer-generated objects by displaying 3D virtual objects registered in a user's natural environment. Figure 1 illustrates an application of this powerful visualization tool where a user can visualize an as-yet unbuilt building in its proposed natural setting. Other applications include interactive 3D illustrations for constructing and for maintaining complex machinery [Feiner, MacIntyre, Seligmann 92, 93] [Caudell, Mizell 92] and in-patient visualization of medical data, e.g., ultrasound [Bajura, Fuchs, Ohbuchi 92]. In all these applications it is vitally necessary for computer-generated objects and real-world objects to be visually registered with respect to each other in every image the user sees. If accurate registration is not maintained, the computer-generated objects appear to float around in the user's natural environment without having a specific 3D spatial position.

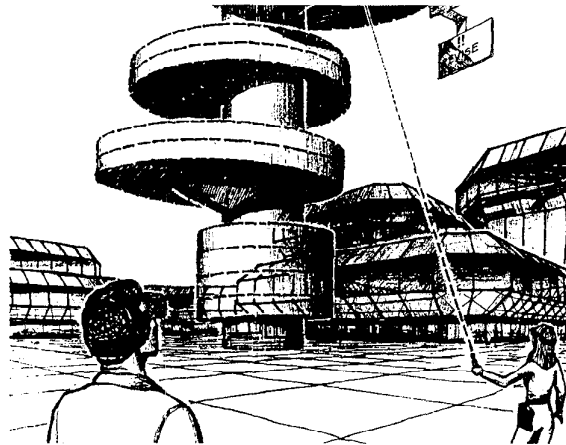


Figure 1: Example augmented-reality application: Visualization of a proposed building design.

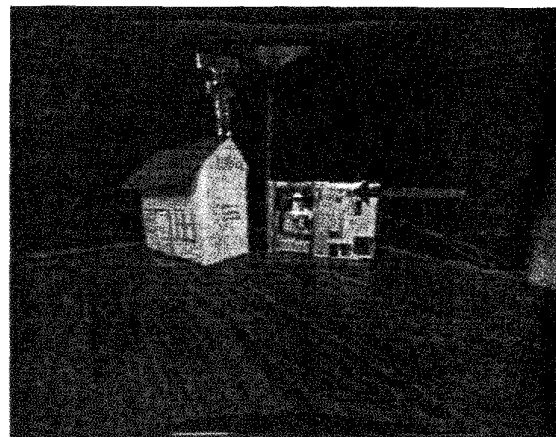


Figure 2: Experimental augmented-reality system showing dynamic registration of a virtual antenna and an annotation arrow which appear to be "nailed" in place.

Figure 2 is an image from our experimental AR system which dynamically corrects image registration on a frame-by-frame basis. It shows a computer-generated television antenna registered correctly on a toy house and a direction

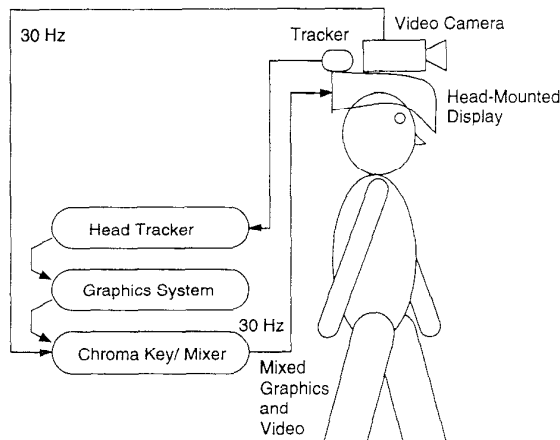


Figure 3: Typical video-based augmented-reality system components.

arrow registered correctly on a disk drive. The antenna and the arrow maintain correct registration in every image the user sees, including when the user is moving. The rest of this paper explains how this result is achieved and suggests future directions for dynamic registration correction. Section 2 describes the current model for augmented-reality systems and the sources of error in them. Section 3 explains a method for dynamically correcting registration error. Section 4 describes the implementation and results of our experimental AR system which dynamically corrects registration error. Conclusions and future directions follow in section 5.

## 2. CURRENT MODEL FOR AR SYSTEMS

Video-based augmented-reality systems are currently based on the model shown in figure 3. The user wears a head-mounted display (HMD) which presents combined images of both real and virtual (computer-generated) objects. Images of real objects are obtained from a video camera mounted on the user's display helmet. Images of virtual objects are generated by a graphics system. A tracking system reports the user's head position to the graphics system so it can render images with a virtual camera model of the video (real) camera's view of the world. The real and virtual images are typically merged with a chroma-key or video mixer for display in the HMD. For clarity in this paper, we discuss only the monocular case with one video camera mounted on the HMD. A stereo system would add a second video camera which would be treated as a second independent monocular system. Constructing a binocular HMD which presents correct stereopsis is problem addressed in [Edwards, Rolland, Keller 93].

The apparent registration between real and virtual objects depends on how accurately the virtual camera models the real one. Figure 4 shows a detailed transformation model for the virtual camera. Virtual objects are positioned by an *Origin-to-Object* transformation which specifies the position and orientation of a virtual object relative to a coordinate system origin. The virtual camera is positioned relative to the coordinate system origin by the composition of two transformations: *Origin-to-Head*, and *Head-to-Camera*. The *Origin-to-Head* transformation is reported by the track-

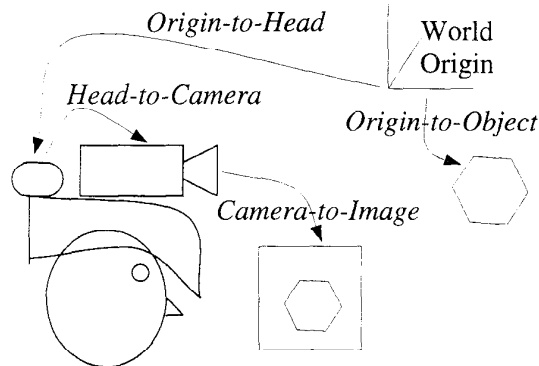


Figure 4: Transformations from object to image.

ing system and specifies the location of a tracking element's position on the user's HMD. The fixed transformation from this tracking element's position to the effective viewpoint of the real camera is the *Head-to-Camera* transformation. Virtual camera images are produced by a perspective projection onto a virtual image plane. A non-linear *Camera-to-Image* mapping is then applied which matches the field of view and lens distortion of the real camera. Typically the *Head-to-Camera* transformation and the *Camera-to-Image* mapping are determined by a calibration procedure such as the one described in section 4.2. It should be noted that this model does not address the problem of correcting for distortion in the HMD optics which is a separate problem from generating correctly registered images [Edwards, Rolland, Keller 93].

Image registration error in combined real and virtual images is caused by the following types of errors:

- 1) The tracking system's origin is not aligned with the world coordinate system origin. This error causes all virtual objects to appear to be displaced from their proper positions.
- 2) The virtual *Origin-to-Object* transformation is not the same as the real *Origin-to-Object* transformation for a particular object. This error causes individual objects to appear out of position.
- 3) The virtual camera position is not the same as the real camera position. This can be caused by errors in either the static *Head-to-Camera* transformation or the dynamic *Origin-to-Head* transformation reported by the tracking system. The tracking system exhibits two types of error: temporal error, and position and orientation error. Position and orientation errors cause misregistration in all cases, while temporal errors cause misregistration only during user movement. Temporal errors are caused by a delay in sensing and reporting tracking information to the computer graphics system and the computer graphics system's delay in generating the appropriate virtual images [Adelstein, Johnston, Ellis 92].
- 4) The virtual *Camera-to-Image* mapping doesn't accurately model the real camera. The *Camera-to-Image* mapping abstraction is that any real camera can be modelled by an idealized pinhole camera with a particular center of projection, viewing direction, field of view, and distortion function. The distortion function is a 2D warp which accounts for the non-linearities found in lens-based projection systems. Errors in the *Camera-to-Image* mapping cause misregistration to vary with screen position.

### 3. CORRECTING REGISTRATION ERROR

The new idea presented here is to dynamically measure the registration error, or misregistration, in each combined image and use that information to correct the system errors that caused the misregistration. In the above model (figure 3), absolute correctness of all the transformations shown in figure 4 is needed for absolute image registration. This situation is much like the design of an "open loop" system from systems theory. An input generates an output which has errors. The only way to improve the system is to make each system component more accurate. Another idea from systems theory is a "closed loop" system where a system's error output is used again at the input to improve the system's output. The model advanced here resembles the "closed loop" design where the image registration error, or misregistration, is used to correct the transformation parameters which caused it. The type of correction which can be performed depends on two main factors: 1) the method used for detecting and measuring image misregistration, and 2) the uncertainty and image-space sensitivity of the different parameters to be adjusted. Both are described below.

Different methods for measuring image misregistration dictate what kinds of correction can be performed. One way to measure image misregistration is to identify a recognizable point on each object to be registered. The image coordinates of each point are located in both the real and uncorrected virtual images. The difference between each point's position in each real image and corresponding uncorrected virtual image is the registration error, or misregistration, for the object corresponding with that point. This measure of misregistration can correct for errors such as camera orientation and sometimes camera position. A drawback with this measure is that neither the distance between an object and the camera nor an object's orientation can be estimated. Another way to measure image misregistration is to attempt to recognize an object's position, size, and orientation in each real image. This information can correct camera to object distance as well as relative object orientation.

Even if a particular misregistration measure does not allow estimation of a particular transformation parameter, misregistration can still be reduced. In many cases parameters which cannot be estimated can be assumed to be correct and the image registration error can be reduced by adjusting the remaining parameters. In other cases there is no way to separate the error contributions from different parameters and one or more must be adjusted depending on their relative uncertainty. The important point is that image registration error can be reduced even if some approximations are made.

The selection of which parameters to adjust depends on both their uncertainty and how sensitive image-space errors are to that uncertainty. For example, if the positions of objects are well known but the camera position and orientation are relatively uncertain, the camera position and orientation should be adjusted instead of object positions. Another example is camera position versus camera orientation. When an object is relatively close to a camera, its projection in image coordinates is more sensitive to the camera's position and less so to its orientation. For objects relatively far from a camera, the camera's orientation most strongly influences where an

object's image appears.

This approach to correcting registration error can also be used to correct temporal errors. If the tracking system delay is longer than the delay in measuring image misregistration, the misregistration can be used to improve the most recent tracking system estimate. In video-based AR systems it is possible to effectively reduce the delay in measuring image registration to zero by delaying the the real video image stream by the time it takes to measure image registration and generate corrected virtual images. This makes it possible to correct temporal and spatial image registration exactly in every image the AR user sees. If there is registration error, it is only because the error compensation algorithm failed. For applications which can tolerate minimal delays, potentially perfect registration can be achieved. This trade-off is not possible with optically based AR systems which allow the user to see his surroundings directly.

Some success at improving registration error has been achieved with autocalibration approaches [Gottschalk, Hughes 93] and predictive tracking techniques [Azuma 94; List 84] which use a state estimate to help predict current measurements. However these approaches still suffer from the "open loop" requirement for perfect tracking and calibration.

### 4. THE EXPERIMENTAL SYSTEM

This section describes an experimental AR system which corrects image registration error on a frame-by-frame basis. Section 4.1 describes the functional components of the system and what hardware is used for them. System calibration is discussed in section 4.2. Section 4.3 describes how registration error is corrected in the experimental system. Section 4.4 shows the results of operating the system both with and without dynamic registration correction.

#### 4.1 System Components

Figure 5 is a schematic for the experimental AR system. It is similar to the system described in figure 3 except for the addition of a real-time video delay and unwarp pipeline and a real-time image feature tracker. The delay and unwarp pipeline delays video by a constant number of frames and optionally applies an inverse distortion function which converts the incoming signal into an equivalent pinhole camera image. With our hardware, it is more practical to undistort the real camera video images to match the undistorted virtual images instead of distorting the virtual images to match the real camera ones. The pipeline delay is adjustable but constant during operation. The pipeline delay is set to match the delay in generating the correct virtual image to mix with the corresponding real camera video frame.

The image feature tracker recognizes features in the real video stream and passes their image coordinates to the graphics system. The features to be detected are red LEDs driven by a 9V power supply. The LEDs are significantly brighter than other objects in the environment. The LEDs are detected by applying a brightness and image area threshold to each image. Correspondence between LEDs and the particular features they represent is established by matching detected LED positions with the nearest estimated feature positions in

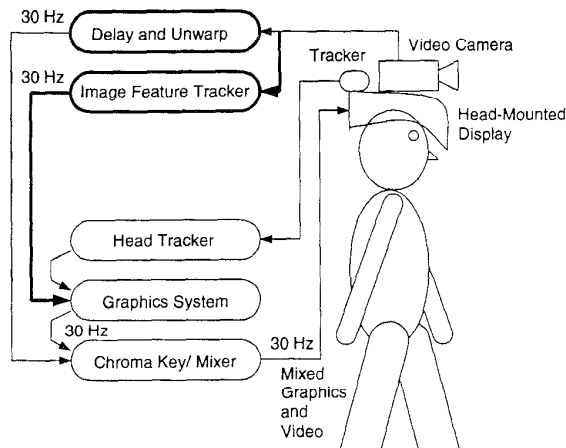


Figure 5: Video-based augmented-reality system model with registration correction.

each corresponding uncorrected virtual image. To do this it is not necessary to render uncorrected virtual images. Only the feature positions must be computed. Once feature correspondence is established, the difference between each feature's position in each real image and its estimated position in each corresponding virtual image can be used to render virtual images which are better registered with the real video images. The methods used for correcting registration are explained in section 4.3.

The camera used this experiment is a *Panasonic GP-KS102* color CCD camera with a highly distorting 110 degree wide angle lens. The head tracking system is an Ascension *A Flock of Birds* magnetic tracking system. The delay and unwarp, image feature tracker, and graphics system are different software modules which utilize separate portions of the *Pixel-Planes 5* graphics multicomputer at UNC [Fuchs 89]. Video is input to the *Pixel-Planes 5* system via a real-time video digitizer and output via a standard double-buffered frame buffer. Although it is desirable to mix the real camera and virtual camera video signals digitally, the bandwidth requirements of 30 Hz operation require the use of an analog Sony *CRK-2000 Universal Chroma Keyer* video mixer.

The AR world of the experimental system consists of a virtual TV antenna positioned atop a real model house (where an LED is located) and a virtual arrow which indicates an adjustment screw on a real disk drive (also where an LED is located).

#### 4.2 Calibration

Before the system can be operated, the *Head-to-Camera* transformation and the *Camera-to-Image* mapping must be estimated. This is done by operating the AR system and using manual feedback to converge on a solution. Optional compensation for non-linear lens distortion in the *Camera-to-Image* mapping is measured by examining a distorted camera image and finding a 2D warp function which converts that image into an undistorted one [Bajura 93]. If non-linear lens distortion is not considered, a best-fit calibration solution by matching field of view is possible even

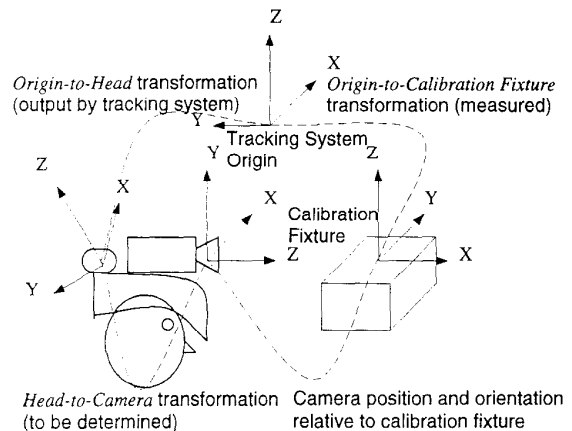


Figure 6: Calibration transformations.

for distorting lenses.

Figure 6 shows how the *Head-to-Camera* transformation is initially estimated. A calibration fixture is used to represent a fixed position and orientation which are measured relative to the tracking system origin. When the camera is placed in a specific position and orientation relative to the calibration fixture, the position and orientation of the head tracking element is recorded. The *Head-to-Camera* transformation is the difference between the head tracking element's position and orientation and the camera's position and orientation. A calibration fixture is needed because the tracking system reports positions relative to a fixed but not precisely known origin. If the tracking system reported coordinates in a known coordinate system relative to itself a calibration fixture wouldn't be necessary.

The calibration fixture is located by using the head tracking element to perform rigid body rotations about each of the calibration fixture's coordinate axes. As rotations about each axis are performed, the tracking element's position and orientation are recorded and used to compute each axis of rotation. Because rigid body rotations are used it isn't necessary to know the offset of the tracking element from each axis of rotation beforehand. The position and orientation of the calibration fixture's coordinate system are determined once rotations about two axes are performed. By taking enough careful measurements it is possible to locate the calibration fixture to nearly the precision of the tracking system itself.

Ideally, only one measurement of the head tracking element is needed to estimate the *Head-to-Camera* transformation while the camera is simultaneously placed in both a specific position and a specific orientation relative to the calibration fixture. Because it is difficult to accurately both position and orient the camera at the same time, separate measurements are made to estimate the position and orientation components of the *Head-to-Camera* transformation.

The *Head-to-Camera* estimation is refined by making further measurements while running the AR system. Virtual 3D coordinate axes are placed at the same position as the calibra-

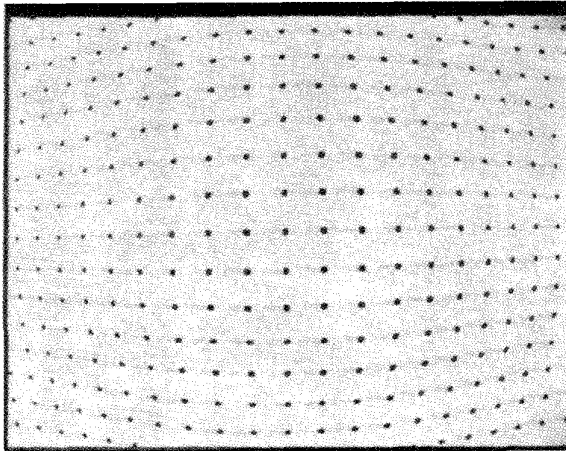


Figure 7: Distorted image of calibration pattern.

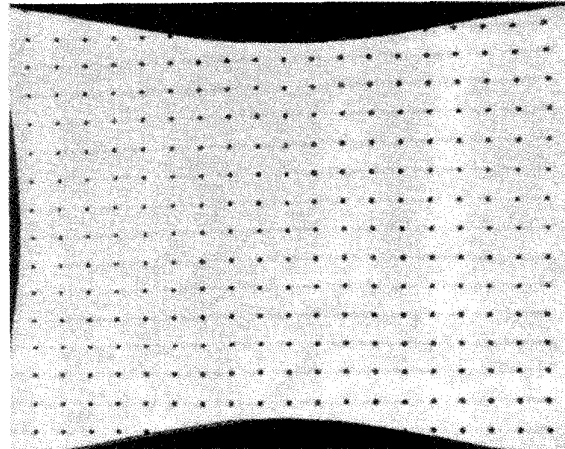


Figure 8: Corrected image of calibration pattern.

tion fixture's coordinate axes determined above. If the *Head-to-Camera* transformation is correct, the virtual and real coordinate axes will appear in the composite images to be aligned in both position and orientation when viewed along the optical axis of the camera. If the coordinate axes are not aligned, the *Head-to-Camera* estimate can be improved with the following heuristics. Only rough estimates for the *Camera-to-Image* mapping are needed at this point because alignment along the optical axis isn't affected by either the field of view or the lens distortion model.

- If the camera is positioned relatively far from the coordinate axes, misregistration is primarily due to *orientation* errors in the *Head-to-Camera* transformation. Rotations about the X and Y axes of the camera orientation should be made to align the axis positions. Rotations about the camera Z axis should be made to align the axis orientations.

- If the camera is positioned relatively near to the coordinate axes, misregistration is primarily due to *position* errors in the *Head-to-Camera* transformation. Translations along the camera X and Y axes should be made to align the axis positions. Translations along the camera Z axis will move the camera viewpoint either in front of or behind a virtual point, for example the coordinate axis origin, when the camera is very close to that point.

Once the *Head-to-Camera* transformation has been adjusted so that the virtual and real coordinate axes appear to be aligned when viewed along the camera's optical axis, the field of view component of the *Camera-to-Image* mapping can be adjusted. This is done by viewing the coordinate axes at angles off the camera's optical axis and separately adjusting the camera's X and Y fields of view until the coordinate axes are realigned. Without lens distortion correction, alignment will not be possible for all off-axis viewing angles. However, misalignment may be minimal with lower distortion lenses.

Non-linear lens distortion in the *Camera-to-Image* mapping is calibrated by imaging a test pattern and finding a distortion function which undistorts the test pattern image. This is

done by appealing to a basic rule of (linear) projective geometry: straight lines remain straight under projection. Scales may change and parallel lines may intersect, but the image of a straight line is always straight. If there is a mapping which converts images from a distorting camera into ones where all straight lines appear to be straight, then the distorting camera can be modelled by a composition of this mapping and a pinhole camera model.

Figures 7 and 8 are images of a test pattern imaged with the 110 degree wide angle camera lens. Figure 7 is the distorted image output from the camera. Figure 8 is a corrected version of the same image. The correction is a radial distortion at the image center which accounts for most of the image distortion [Weng, Cohen, Herniou 92].

The important point about calibration is that it is difficult to do accurately, particularly when the tracking system used has noticeable tracking error throughout its working volume. Using a more accurate measuring device to measure the *Head-to-Camera* transformation would not eliminate errors in the AR system because camera position would still be a function of the tracking system which reports the *Origin-to-Head* transformation [Janin, Mizell, Caudell 93].

#### 4.3 Correcting Registration Error

The image registration model of matching a point on each object makes it difficult to determine which particular errors are causing misregistration. One way to think about this is to consider the misregistration as a function of the camera position and orientation error (a composition of errors in the *Origin-to-Head* and *Head-to-Camera* transformations), *Camera-to-Image* mapping error, and *Origin-to-Object* transformation error:

$$\text{Misregistration} = f(\text{camera position and orientation error,} \\ \text{Camera-to-Image mapping error,} \\ \text{Origin-to-Object transformation error})$$

Misregistration can be reduced by modifying one or more of the parameters which might be causing it. Two approaches to reducing registration error are studied in this experiment.

One approach assumes that the camera position and orientation are absolutely correct and that misregistration is due to errors in the *Camera-to-Image* mapping and *Origin-to-Object* transformation. The second approach assumes that the *Camera-to-Image* mapping and *Origin-to-Object* transformations are correct and that the camera position and orientation are in error. Neither of these approaches is optimal in the sense of minimizing error by smoothly adjusting all the possible parameters according to parameter certainty and registration sensitivity, e.g. optimal filtering. Such an analysis is difficult to make and may not be any better than making a few reasonable assumptions. Both of the approaches tried here are relatively easy to implement and are sensible in certain situations.

In the first correction approach, if the reported position and orientation of the virtual camera are assumed to be correct, there is no way to tell whether registration errors were caused by incorrect *Camera-to-Image* mapping, incorrect *Origin-to-Object* transformations, or both. By making a further assumption that the *Camera-to-Image* mapping is also correct, object positions alone can be adjusted to account for any registration error. To render a corrected image, each misregistered object is temporarily displaced to a position where it will appear to be registered correctly. This correction produces combined images with no measured registration error. Since the registration metric gives no estimate of distance between each object and the camera, virtual objects are displaced on a constant radius (rotated) from the virtual camera viewpoint. This maintains the best estimate of distance between the camera and each object so that objects don't grow and shrink unnaturally.

In the second correction approach, the virtual camera viewpoint is corrected to reduce registration error while object position and camera distortion are assumed to be correct. If enough features are visible it is theoretically possible to compute both camera position and orientation from the 3D (X,Y,Z) feature positions and their corresponding (U,V) image locations. If the feature positions aren't degenerate, the camera position and orientation can be recovered by non-linear methods with a minimum of 4 points and by linear methods with a minimum of 6 points [Horaud, Conio, Lebloulloux 89] [Ganapathy 84]. Trying to correct the camera position this way isn't practical for at least three reasons. First, there is no way to guarantee enough features will be visible in every image. Second, these solution methods are highly sensitive to noise and spatial feature distribution. Third, a good estimate of the virtual camera position is already available.

The easiest simplification to make is that the virtual camera position is correct as reported by the *Origin-to-Head* and *Head-to-Camera* transformations and that the registration error is entirely due to camera orientation error. This is a good assumption for three reasons. First, orientation corrections can be made when only one feature is visible. If more than one feature is visible a best-fit solution can be found. Second, under the assumption that objects are relatively far from the camera, which is true in most AR applications, registration errors are much more sensitive to errors in camera orientation than camera position. This means that solving for

camera position is unstable (sensitive to errors) and that solving for camera orientation (when camera position is fixed) is well-behaved (relatively insensitive to errors). Third, tracking system data has more error in rotation than in translation. This is because HMD wearers typically rotate their heads faster than they move them and the head tracking system used incurs significant delays in reporting measurements (temporal error) [Liang, Shaw, Green 91]. In the experimental system, camera orientation error is adjusted by considering only one "reference" feature position and rotating the virtual camera to align that position. This is only an approximation which can correct the alignment of a particular point but not an orientation about that point.

#### 4.4 Registration Results

The experimental system (figure 6) can be operated in nine different modes by different selections of the two parameters *real-video-delay* and *registration-correction-method*. *Real-video-delay* is one of:

- 1) no delay or distortion correction
- 2) delay without distortion correction
- 3) delay with distortion correction.

*Registration-correction-method* is one of:

- A) none
- B) correction by adjusting *Camera-to-Image* mapping and/or *Origin-to-Object* transformations (move the object)
- C) correction by adjusting camera orientation (rotate the camera).

The results of different combinations of these parameters are described below:

(1,A): This "open loop" mode is equivalent to the "current model" shown in figure 3. Figure 9 shows the result: the virtual objects are not aligned with their proper positions and lag noticeably behind during user movement in spite of careful calibration and system tuning.

(1,B): This option has good registration at the object feature positions except during user motion when the registration still lags noticeably. It appears to be possible to shake the virtual objects from their proper positions, but they always return. This case shows the simple power of the "closed loop" system model over the "open loop" system model in figures 3 and 9. Despite the lack of lens distortion correction, noticeable lag, and various other errors, the virtual objects still appear to belong in specific spatial positions, a result not easily achieved without dynamic registration correction.

(2,A), (2,B): These combinations have the same static results as (1,A) and (1,B) above. However the registration error during motion (temporal registration error) is extremely small because the real video delay is the same as the tracking and image generation delay – the dynamic registration appears to be the same as the static registration. The reduction in the "swimming" of the virtual objects during motion makes them appear much more stationary and solid, even in the case of (2,A) where the registration is poor.

(3,A): The addition of lens distortion correction without registration correction produces the best "open loop" operation

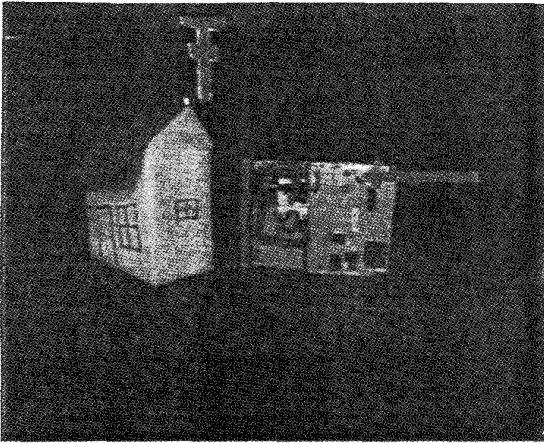


Figure 9: "Open loop" mode without dynamic registration correction or distortion correction. Virtual objects "swim" around and are poorly registered.

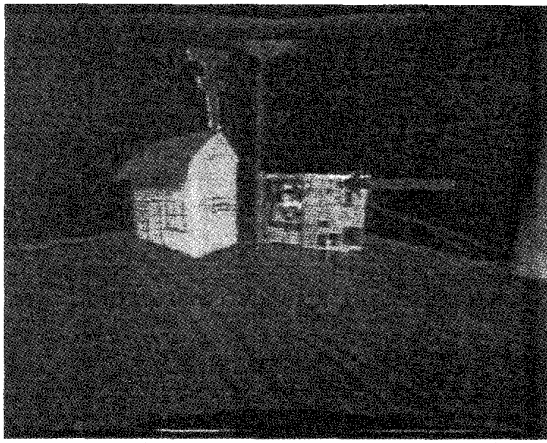


Figure 11: "Closed loop" mode with optical distortion correction, video delay, and dynamic correction of object position and lens distortion. Virtual objects appear to be "nailed" to their reference positions.

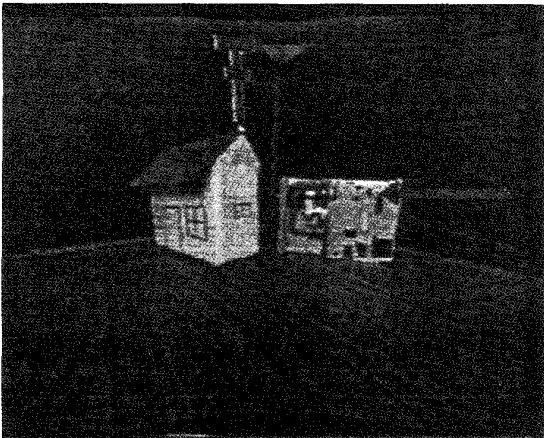


Figure 10: "Open loop" mode with video delay and optical distortion correction. Virtual objects do not swim as much and registration is somewhat improved.

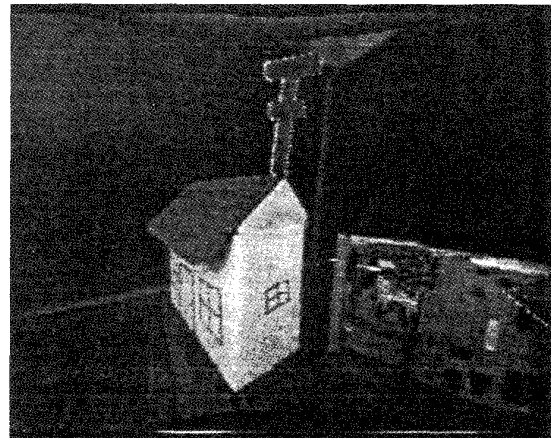


Figure 12: "Closed loop" mode with optical distortion correction, video delay, and camera orientation correction to align base of antenna. Adjustment arrow is steady but slightly misregistered.

possible with the experimental system (figure 10). The lens distortion correction improves registration considerably but the virtual objects still wander slightly during movement and appear in different positions as the tracking system exhibits errors within its working volume.

(3,B): This combination of distortion correction, delay, and registration correction by displacing objects produces the best registration in the experimental system (figure 11). In all cases, during both static and dynamic viewing, the virtual objects appear to be registered correctly with respect to their reference positions. They appear to be "nailed" in place.

(3,C): Here only the reference position for the TV antenna is used to adjust the virtual camera orientation while the real video is corrected for distortion and delayed (figure 12). No registration correction is made for lens distortion or object position errors. This combination produces the 2nd best registration after combination (3,B). The base of the antenna

appears to be registered correctly on the house, but the arrow on the disk drive adjustment screw consistently appears to be just a bit low. This misregistration could be caused by errors in the *Origin-to-Object* transformations for the TV antenna and disk drive screw or by errors in the initial camera orientation which aren't completely corrected with this method.

(1,C), (2,C): These combinations did not make sense. Without lens distortion correction it is not possible to modify the camera position to improve registration for more than one object.



## 5. CONCLUSIONS AND FUTURE DIRECTIONS

- Building augmented-reality systems with accurate registration is difficult. The visual registration requirement between real objects and virtual objects exposes any measurement or calibration error in an AR system. The strongest argument in favor of dynamic compensation is that no matter how much measurement and calibration are performed, there may (will) still be errors in the composite images. What the real camera sees must be taken as the ground truth and the registration error between the real and virtual images must be corrected by compensating for errors in calibration, tracking, and/or distortion correction. It is more practical to measure and correct errors using a "closed loop" design than to avoid making them in the first place with an "open loop" design.

- The experiment described here demonstrates the importance and feasibility of dynamically measuring and correcting image space registration error. The experimental system is more stable and better aligned than systems without registration correction.

- The idea of measuring and correcting image registration error has implications for the design of future augmented-reality systems. Since feedback can compensate for tracking errors, in essence becoming part of the tracking system itself, less accurate and less expensive tracking systems may be feasible. Optical tracking systems [Azuma 94] could be designed to use stationary cameras to track a user's position while cameras on the user's head could look outward to determine the user's orientation. Feedback also reduces the accuracy requirements for lens distortion correction and system calibration.

- The success of registration correction depends on the ability to accurately measure registration in the first place. This is not a simple task in general. The experiment described here uses an oversimplified method for measuring registration which may not be practical in many environments. A large amount of work in this area has already been done by the computer vision community. Hopefully some of their results can be applied to AR systems.

- Correct occlusion cues are still needed for augmented-reality systems to be truly believable. This method of registration only works for virtual objects which are completely in front of real ones. What is really needed is a way to sense positions and depths in the environment from the real camera. With such information, the reference positions could be used to position virtual objects which could be hidden properly if they were obscured.

## 6. ACKNOWLEDGMENTS

Support for this research was provided by The Link Foundation and the NSF/ARPA Science and Technology Center for Computer Graphics and Scientific Visualization (NSF Cooperative Agreement #ASC8920219). Thanks to: Andrei State for Figure 1, Henry Fuchs for writing suggestions.

## 7. REFERENCES

[Adelstein, Johnston, Ellis 92] Adelstein, B., Johnston, E., Ellis, S. "A testbed for Characterizing Dynamic Response of Virtual Environment Spatial Sensors," Proceedings 5th Annual ACM Symposium on User

Interface Software and Technology (UIST) 1992, ACM SIGGRAPH/SIGCHI, Monterey, California, Nov. 1992, pp.15-22.

[Azuma 94] Azuma, R., Bishop, G. "Improving Static and Dynamic Registration in an Optical See-through HMD," Computer Graphics (Proceedings of Siggraph 1994), pp.197-204.

[Bajura 93] Bajura, M. "Camera Calibration for Video See-Through Head-Mounted Display," TR93-048 Computer Science Technical Report UNC Chapel Hill, July 1993

[Bajura, Fuchs, Ohbuchi 92] Bajura, M., Fuchs, H., Ohbuchi, R. "Merging Virtual Reality with the Real World: Seeing Ultrasound Imagery within the Patient," Computer Graphics (Proceedings of Siggraph 1992), pp.203-210.

[Caudell, Mizell 92] Caudell, T.P., Mizell, D.W. "Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes," Proceedings Hawaii Intl. Conf. on System Sciences, Jan 1992, vol. 2, pp.659-669.

[Edwards, Rolland, Keller 93] Edwards, E.K., Rolland, J.P., Keller, K.P. "Video See-through Design for Merging of Real and Virtual Environments," Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS) 1993, Seattle, WA.

[Feiner, MacIntyre, Seligmann 92] Feiner, S., MacIntyre, B., Seligmann, D. "Annotating the Real World with Knowledge-Based Graphics on a See-Through Head-Mounted Display," Proceedings Graphics Interface 1992, Canadian Information Proc. Soc., pp.78-85.

[Feiner, MacIntyre, Seligmann 93] Feiner, S., MacIntyre, B., Seligmann, D. "Knowledge-Based Augmented Reality," Communications of the ACM, July 1993. Vol. 30, No. 7, pp.53-62.

[Ganapathy 84] Ganapathy "Real-Time Motion Tracking Using a Single Camera," AT&T Bell Labs Tech Report 11358-841105-21-TM.

[Gottschalk, Hughes 93] Gottschalk, S., Hughes, J. "Autocalibration for Virtual Environments Tracking Hardware," Computer Graphics (Proceedings of Siggraph 1993), pp.65-72.

[Horand, Conio, Lebouilleux 89] Horand, R., Conio, B., Lebouilleux, O. "An Analytic Solution for the Perspective 4-Point Problem," Computer Vision, Graphics, and Image Processing, Vol 47, No. 33-34 (1989), pp.33-43.

[Janin, Mizell, Caudell 93] Janin, A.L., Mizell, D.W., Caudell, T.P. "Calibration of Head-Mounted Displays for Augmented Reality Applications," Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS) 1993, Seattle, WA.

[List 84] List "Nonlinear Prediction of Head Movements for Helmet-Mounted Displays," Tech. report AFHRL-TP-83-45, Williams AFB, AZ.

[Liang, Shaw, Green 91] Liang, J., Shaw, C., Green, M. "On Temporal-Spatial Realism in the Virtual Reality Environment," Proceedings 4th Annual ACM Symposium on User Interface Software and Technology (UIST) 1991, ACM SIGGRAPH/SIGCHI, Hilton Head, South Carolina, Nov. 1991, pp.19-25.

[Weng, Cohen, Herniou 92] Weng, J., Cohen, P., Herniou, M. "Camera Calibration with Distortion Models and Accuracy Evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 14, No. 10, October 1992, pp.965-980.