

# Augmenting Reality Using Affine Object Representations

Dr. James Vallino  
Department of Computer Science  
Rochester Institute of Technology  
{jrv@cs.rit.edu}

Dr. Kiriakos N. Kutulakos  
Department of Computer Science  
University of Rochester  
{kyros@cs.rochester.edu}

## 1. Introduction

An augmented reality system is a system that creates a view of a real scene that visually incorporates into the scene computer-generated images of three-dimensional virtual objects. As the user of such a system moves about the real scene the virtual objects appear as if they actually exist in the scene. One motivation for augmenting reality in this way is to enhance the performance of real world tasks. The performance requirements for an augmented reality system are: (1) merge images of 3D virtual objects with images of the real environment, (2) generate a consistent view of those objects from all views of the real scene, and (3) perform these operations in real-time to be interactive with the user. Augmented reality can be compared to the more commonly known virtual reality. Virtual reality systems immerse a user in an environment that is completely computer-generated. Augmented reality systems, on the other hand, strive to maintain the user's immersion in the real environment. The rationale behind this is twofold. First, real environments contain a wealth of information much of which is impossible to model and simulate by computer. Secondly, if the end goal is to enhance the performance of a real-world task the user will most naturally perform that task while looking at an augmented view of the real scene. Practical applications for augmented reality are described in other chapters of this book and include applications from the domains of manufacturing (Chapter ????) Baird

and Ioannou), medicine (Chapter ??? Satava), and the military (Chapter ??? Tappert).

Both virtual reality and augmented reality systems provide an interface that allows the user to operate in a natural 3D physical space while receiving a consistent set of sensory inputs for both the real and virtual worlds. The primary performance goal for a virtual reality system is to present visual stimuli that are consistent with the changes in body position sensed by the user. Any inconsistency perceived by the user results from a misregistration between the coordinate system the user is maintaining internally to describe body position and the coordinate system that describes the graphics system's viewpoint in the virtual scene. This can be contrasted to the primary performance goal for an augmented reality system which is to render views of virtual objects that are consistent with the user's view of the real environment containing the objects. Any inconsistency, which manifests itself as a difference between two visual stimuli, i.e. the virtual and real images, derives from a misregistration between the coordinate system describing the user's viewpoint in the real scene and the graphics system's viewpoint in the virtual scene (Chapter ??? Rolland, ??? Azuma). The nature of this registration problem in augmented reality systems can be seen in Figure 1a. To create an image of the three-dimensional virtual objects that is consistent with the user's current view of the world and the object's placement in the real world requires the definition of the geometric relationships between the virtual and physical objects shown in Figure 1a. Any errors in the determination of these relationships appear to the user as inconsistencies in the appearance of the virtual objects in the real scene (Figure 1b,c). These errors in registering the two images, are classified as either static or dynamic (Chapter ??? Holloway). Static errors are perceived by the user as differences in the placement or appearance of the virtual objects when viewed from different viewpoints. The dynamic errors are caused by the system lagging behind due to not meeting its real-time requirements. The visual effect of these dynamic errors is a shift in the position of the virtual objects when there is motion in the system.

### **Desired position for Figure 1.**

This chapter describes a method for solving the registration problem in augmented reality systems using affine object representations. The method defines a global non-Euclidean affine coordinate system and determines the relationships between that global coordinate system and all the coordinate systems in Figure 1a. Unlike other solutions to the augmented reality registration problem that require position sensing and calibrated cameras, this method relies solely on tracking four or more features in video images of the real scene using uncalibrated cameras. As shown in Figure 1a, our approach requires that a video camera view the real scene. This requirement favors operation with a monitor-based display or video see-through head-mounted display (Azuma 1997). This chapter describes working augmented reality systems that employ both those display types. Operation with an optical see-through display, in addition to requiring the video camera, will require alignment of the camera with the see-through display (Janin, Mizell et al. 1993; Hoff, Nguyen et al. 1996).

## **2. The Registration Problem**

The key requirement for creating an augmented reality image in which virtual objects appear to exist in the three dimensional real scene is knowledge of the relationships between the object, world, and camera coordinate systems (Figure 1a). These relationships are determined by the object-to-world,  $\mathbf{O}$ , world-to-camera,  $\mathbf{C}$ , and camera-to-image plane,  $\mathbf{P}$ , transforms. The object-to-world transform specifies the position and orientation of a virtual object with respect to the world coordinate system that defines the real scene. The pose of the video camera that views the real scene is defined by the world-to-camera transform. The projection performed by the camera to create a 2D image of the 3D real scene is specified in the camera-to-image plane transform. Visually correct merging of virtual objects with the live video image requires

computation of these relationships. Accurately performing this computation while maintaining real-time response and a low latency is the major challenge for an augmented reality system.

## 2.1 Augmenting reality using pose sensors

In many augmented reality systems, the problem of computing the transforms shown in Figure 1a is approached in a straightforward manner using sensing, calibration and measurement to explicitly determine each transform (Feiner, MacIntyre et al. 1993; Ahlers, Breen et al. 1994; State, Chen et al. 1994). Sensors, based on mechanical, magnetic or optical techniques, are used to measure the position and angular orientation of the camera with respect to the world coordinate system. These two measurements together are termed the *pose* of the camera and determine the world-to-camera transform, **C**. Quantifying the camera-to-image transform, **P**, requires knowledge of the intrinsic parameters, such as focal length and aspect ratio, of the camera. These can be determined by performing a calibration procedure on the camera (Tsai 1987). The third transform, **O**, is computed by simple measurement. The world coordinate system is a standard three-dimensional Euclidean space. The desired position and orientation for a virtual object can be measured in the real scene. Using the methods just described, all of the necessary transforms are known so that, at least in principle, virtual objects can be rendered and merged correctly with the live video.

These approaches do suffer from limitations. So far none of the pose sensors have been completely satisfactory for an augmented reality application (Azuma 1997). Mechanical sensors place limits on the range of the work space and require attachment to a restrictive linkage. Magnetic sensors are susceptible to disturbances in their generated magnetic field created by metal objects in the workspace. Calibration for these distortions can reduce the errors (Byrson 1992). The magnetic sensors also have latencies that can only be improved with predictive

estimation of pose (Azuma and Bishop 1994). Techniques for calibrating a camera to determine its intrinsic parameters are available (Tsai 1987) but it is a tedious process to perform. The intrinsic parameters of a camera may change over time requiring recalibration. In particular, zoom lenses, like those found on common consumer-grade video cameras, may change focal length with use either intentionally or with wear. Accurate sensing of zoom position is not commonly available which would require recalibration with each change in focal length. Any errors introduced by incorrect pose sensing or camera calibration propagate through the system and will appear as misregistration in the final augmented reality image. The optical see-through head-mounted displays used in some augmented reality systems also must be calibrated even if the system does not use a video camera (Janin, Mizell et al. 1993). Any inaccuracy in the calibration of the display will result in misregistration in the augmented image.

## **2.2 Computer vision for augmented reality**

The initial approaches to augmented reality discussed in the previous section all overlook one source of significant information. Computer vision research has developed techniques for extracting information about the structure of a scene, the intrinsic parameters of the camera, and its pose from images of the scene. Recent augmented reality systems are applying computer vision methods to improve performance. Tuceryan et al. (1995) provides a careful analysis of procedures that rely on computer vision for calibrating a monitor-based augmented reality system. Image analysis of the video signal at runtime can also be beneficial. Several systems have been described in the literature that track fiducials in the scene at runtime. Bajura and Neumann (1995) track LED fiducials to correct registration errors. Other systems (Hoff, Nguyen et al. 1996; Neumann and Cho 1996) use knowledge of the intrinsic camera parameters and tracking of fiducials placed in known locations in the scene to invert the camera projection operation and obtain an estimate of the viewer pose.

A hybrid method which uses fiducial tracking in combination with standard magnetic position tracking (State, Hirota et al. 1996) requires an initialization procedure that determines the intrinsic parameters of the cameras viewing the scene. Fiducials, whose location in the scene are known, are tracked in two video images. The position of the viewer is computed by inverting the projection operation. Position data obtained from a magnetic tracker aides in localization of the landmarks. This aide is particularly useful when large motions are encountered between two video frames. Algorithms that rely solely on vision-based tracking often can not determine the interframe correspondences between fiducials when large motions occur between frames. The magnetic tracker position estimates are also used when occlusions prevent the vision system from seeing the required minimum number of fiducials.

Mellor (1995a; 1995b) and Uenohara and Kanade (1995) describe two augmented reality systems that eliminate the need for explicit determination of the viewer's pose. For each viewpoint, Mellor (1995a; 1995b) uses a linear method to solve for the camera's projection transform from the positions of tracked fiducials. The pose, specifying the 3D position and orientation of the camera, is not directly determined. Instead the computed projection transform is used to render the virtual objects. The camera must be calibrated at initialization time and a laser range finder provides the 3D positions of the fiducials in the scene. The second method for obtaining correct registration with neither position tracking nor camera calibration is presented by Uenohara and Kanade (1995). They track features in live video and represent the virtual points associated with planar overlays as the linear combination of feature points. The placement and rendering of three-dimensional virtual objects were not considered.

### 3. Augmenting Reality using Affine Representations

The approach to augmenting reality that is described in the following sections is motivated by recent computer vision research that has determined structure for objects in a scene and the pose of the camera viewing it without knowledge of the object-to-world, world-to-camera and camera-to-image plane transforms. The following observation was provided by Koenderink and van Doorn (1991) and Ullman and Basri (1991):

*Given a set of four or more non-coplanar 3D points, the projection of all points in the set can be computed as a linear combination of the projection of just four of the points.*

This observation will be used to create a global coordinate system in which the coordinate systems diagrammed in Figure 1a can be expressed. Additionally, this global coordinate system will be defined solely from the locations of visible features in the real scene with no knowledge of the intrinsic parameters and pose of the camera.

#### 3.1 Affine Camera Approximation

Accurate determination of where a point on a virtual object will project in the video image is essential for correct registration of the virtual and live-video images (Foley, van Dam et al. 1990; Tuceryan, Greer et al. 1995). In homogeneous coordinates, the projection,  $[u \ v \ h]^T$ , in the video image of a 3D point  $[x \ y \ z \ w]^T$  can be expressed using the equation:

$$\begin{bmatrix} u \\ v \\ h \end{bmatrix} = \mathbf{P}_{3 \times 4} \mathbf{C}_{4 \times 4} \mathbf{O}_{4 \times 4} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}. \quad (1)$$

The transforms  $\mathbf{P}_{3 \times 4}$ ,  $\mathbf{C}_{4 \times 4}$ , and  $\mathbf{O}_{4 \times 4}$  are shown in Figure 1a and are the camera-to-image

plane, world-to-camera and object-to-world transforms respectively. Equation 1 assumes that object, world and camera coordinate systems are independently defined. Previous approaches to augmented reality have been based on an explicit determination of each of the transforms that relate the coordinate systems. Our approach will represent the three coordinate systems in a single *non-Euclidean* coordinate system and express the projection operation with the equation:

$$\begin{bmatrix} u \\ v \\ h \end{bmatrix} = \Pi_{3 \times 4} \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}, \quad (2)$$

where  $[x' \ y' \ z' \ w']^T$  are the new coordinates for the point transformed from  $[x \ y \ z \ w]^T$ .

To fully exploit this simplification and the observation of Koenderink and van Doorn (1991) and Ullman and Basri (1991) we will use *weak perspective projection* to model the camera-to-image plane transform  $\mathbf{P}$  (Shapiro, Zisserman et al. 1995). Under a weak perspective approximation, the projections in the image plane of 3D points are determined by first projecting the points along parallel rays orthogonal to the image plane. The entire image is then scaled by  $f / z_{avg}$  where  $f$  is the camera's focal length and  $z_{avg}$  is the average distance of the points from the image plane. This approximation is commonly seen in the computer vision literature and holds when the front to back depth of objects along the viewing direction is small compared to the viewing distance (Thompson and Mundy 1987). While this does impose restrictions on the system it has also been shown that the weak perspective approximation can yield more accurate structure-from-motion computations (Boufama, Weinshall et al. 1994; Wiles and Brady 1996).

### 3.2 Global Affine Coordinate System

All points in this system are represented with an *affine representation* whose coordinate system is defined using the location of feature points in the image. This representation is



invariant when an affine transform, i.e. translation, rotation, non-uniform scaling, is applied to all points. Transforms caused by the motion of a weak perspective camera viewing a scene will maintain this affine invariant representation. *Affine reprojection or transfer* (Barrett, Brill et al. 1992; Shashua 1993) is used to compute the projection of virtual objects placed into the real scene.

The affine representation for a collection of points,  $p_0, \dots, p_n$ , is composed of: the *affine basis points* which are four non-coplanar points, one of which is specially designated as the *origin*; and the *affine coordinates* of each point that define the point with respect to the affine basis points. The properties of affine point representation are illustrated in Figure 2.

**Desired location** for Figure 2

Our affine augmented reality systems are based on two properties of affine representations (Koenderink and van Doorn 1991; Mundy and Zisserman 1992; Weinshall and Tomasi 1993):

**Property 1 (Affine Reprojection Property)** *The projection,  $[u_p \ v_p]^T$ , of any point,  $p$ , represented with affine coordinates  $[x \ y \ z]^T$ , is expressed by the equation:*

$$\begin{bmatrix} u_p \\ v_p \end{bmatrix} = \underbrace{\begin{bmatrix} u_{p_1} - u_{p_0} & u_{p_2} - u_{p_0} & u_{p_3} - u_{p_0} \\ v_{p_1} - v_{p_0} & v_{p_2} - v_{p_0} & v_{p_3} - v_{p_0} \end{bmatrix}}_{\Pi_{2 \times 3}} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} u_{p_0} \\ v_{p_0} \end{bmatrix} \quad (3)$$

where  $\begin{bmatrix} u_{p_i} & v_{p_i} \end{bmatrix}^T$ ,  $i = 0, \dots, 3$  are the projections of the origin  $p_0$ , and the three other basis points,  $p_1, p_2$ , and  $p_3$ , that define the affine coordinate system. This can be equivalently expressed in homogeneous coordinates with the equation:

$$\begin{bmatrix} u_p \\ v_p \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} u_{p_1} - u_{p_0} & u_{p_2} - u_{p_0} & u_{p_3} - u_{p_0} & u_{p_0} \\ v_{p_1} - v_{p_0} & v_{p_2} - v_{p_0} & v_{p_3} - v_{p_0} & v_{p_0} \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\Pi_{3 \times 4}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (4)$$

Equation 4 provides an explicit definition for the projection matrix  $\Pi_{3 \times 4}$  seen in Equation 2 and defines the projection of a 3D point in any new image as a linear combination of the projections of the affine basis points in that image. Equation 4 provides a method by which an augmented reality system can calculate the projection of a point on a virtual object with knowledge of only the location of the projections of the affine basis points and the homogeneous affine coordinates for the virtual point. The affine basis points will be defined by visually tracking features in the scene and determining the projections in each new video image. The following property provides the technique for determining the affine coordinates of any 3D point.

**Property 2 (Affine Reconstruction Property)** *The affine coordinates of any point can be computed from Equation 4 if its projection in at least two views is known and the projections of the affine basis points are also known in those views.*

This results in an over-determined system of equations based on Equation 4. Given two views,  $I_1, I_2$ , of a scene in which the projections of the affine basis points,  $p_0, \dots, p_3$ , are known then the affine coordinates  $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$  for any point  $p$  can be recovered from the solution of the following equation:

$$\begin{bmatrix} u_p^1 \\ v_p^1 \\ u_p^2 \\ v_p^2 \end{bmatrix} = \begin{bmatrix} u_{p_1}^1 - u_{p_0}^1 & u_{p_2}^1 - u_{p_0}^1 & u_{p_3}^1 - u_{p_0}^1 & u_{p_0}^1 \\ v_{p_1}^1 - v_{p_0}^1 & v_{p_2}^1 - v_{p_0}^1 & v_{p_3}^1 - v_{p_0}^1 & v_{p_0}^1 \\ \frac{u_{p_1}^2}{2} - \frac{u_{p_0}^2}{2} & \frac{u_{p_2}^2}{2} - \frac{u_{p_0}^2}{2} & \frac{u_{p_3}^2}{2} - \frac{u_{p_0}^2}{2} & \frac{u_{p_0}^2}{2} \\ v_{p_1}^2 - v_{p_0}^2 & v_{p_2}^2 - v_{p_0}^2 & v_{p_3}^2 - v_{p_0}^2 & v_{p_0}^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (5)$$

where  $\begin{bmatrix} u_p^i & v_p^i \end{bmatrix}^T$  and  $\begin{bmatrix} u_{p_j}^i & v_{p_j}^i \end{bmatrix}^T$  are the projections of point  $p$  and affine basis point  $p_j$ , respectively, in image  $I_i$ .

### 3.3 Affine augmented reality

One performance goal for an augmented reality system is the ability to operate in real-time which will often limit the photorealism possible in rendering the virtual objects. As a minimum the very basic operation of hidden surface removal (Foley, van Dam et al. 1990) must be performed to have a correct visualization of any three-dimensional virtual object. The hidden surface removal algorithm uses a front-to-back ordering of surfaces to determine visibility. That ordering is obtained by assigning a depth to each rendered point that represents its distance from the viewpoint. The following two properties will extend the familiar notions of “image plane” and “viewing direction” to the affine representation. This extension will allow for the required front-to-back ordering of virtual surfaces and use of hardware-supported rendering via z-buffering. The image plane and viewing direction define a 3D coordinate system that describes the orientation of the camera. The graphics system can operate entirely within this global affine coordinate system and completely ignore the original object representation.

**Property 3 (Affine Image Plane)** *Let  $\chi$  and  $\psi$  be the homogeneous vectors corresponding to the first and second row of  $\Pi_{2 \times 3}$ , respectively. (1) The vectors  $\chi$  and  $\psi$  are the directions of the rows and columns of the camera, respectively, expressed in the coordinate frame of the affine*

basis points. (2) The affine image plane of the camera is the plane spanned by the vectors  $\chi$  and  $\psi$ .

The unique direction in space along which all points project to a single pixel in the image defines the viewing direction of a camera under our model of weak perspective projection. In the affine case, this direction is expressed mathematically as the null-space of the matrix  $\Pi_{2 \times 3}$ :

**Property 4 (Affine Viewing Direction)** When expressed in the coordinate frame of the affine basis points, the viewing direction,  $\zeta$ , of the camera is given by the cross product

$$\zeta = \chi \times \psi. \quad (6)$$

Property 4 guarantees that the set of points  $\{p + t\zeta, t \in \mathfrak{R}\}$  that defines the line of sight of a point  $p$  will project to the same pixel under Equation 3. The z-buffer value needed for hidden surface removal that is assigned to every point is the dot product  $[\zeta^T \ 0] \cdot p^T$ . The actual magnitude of this value is irrelevant: the important characteristic is that the front-to-back order of virtual points rendered to the same pixel is correctly maintained along the camera viewing direction.

The affine image plane and viewing direction vectors define a 3D coordinate system that in general will not be an orthonormal reference frame in Euclidean 3D space. Despite this, correct visible surface rendering of any point  $[x \ y \ z \ 1]^T$  defined in the global affine coordinate system can be performed by applying the transform:

$$\begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} u_{p1} - u_{p0} & u_{p2} - u_{p0} & u_{p3} - u_{p0} & u_{p0} \\ v_{p1} - v_{p0} & v_{p2} - v_{p0} & v_{p3} - v_{p0} & v_{p0} \\ & \zeta^T & & 0 \\ & \mathbf{0} & & 1 \end{bmatrix}}_{\Pi_{4 \times 4}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (7)$$

where  $u$  and  $v$  are the graphic image coordinates of the point and  $w$  is its assigned z-buffer value.

$\Pi_{4 \times 4}$  has the same form as the viewing matrix that is commonly used in computer graphics systems to perform transforms of graphic objects. The structural similarity allows standard graphics hardware to be used for real-time rendering of objects defined in the affine coordinate system developed here. In our system, a Silicon Graphics Infinite Reality Engine is directly used to render virtual objects with hardware-supported hidden surface removal. The graphics system not only renders the virtual objects, but may also render affine models of real objects, as described in the next section when we discuss occlusions.

### 3.4 Rendering virtual objects

The affine projection matrix,  $\Pi_{4 \times 4}$ , will correctly render virtual objects for merging with the video image provided that those objects are represented in the global affine coordinate system. A method to define and place objects in this global coordinate system is needed. These operations will have to be performed at runtime since the structure of the coordinate system is not known a priori. The interactive methods are based on the *affine reconstruction property*. The resulting transform operations for rendering a virtual object are shown in Figure 3.

**Desired location** for Figure 3

In order to use  $\Pi_{4 \times 4}$  to render the virtual objects, those objects must be represented in the global affine coordinate system defined by the tracked basis points. The 3D object-centered Euclidean coordinate system that will commonly describe a virtual object must be transformed to the global affine coordinate system developed in the previous sections. The calculation of this object-to-affine transform will be done at runtime. In the simplest approach, the user interactively specifies this transform by placing a real object in the scene that defines a bounding

box for the virtual object. In two separate views of the scene the user will specify the locations of four points defining the bounding box. The *affine reconstruction property* is then applied to determine the affine coordinates of the bounding box. The bounding box of the virtual object is used to compute the *object-to-world transform*,  $\mathbf{O}_{4 \times 4}$ , from Figure 1a. In this case the world coordinate system is the common affine coordinate system. The computed transform handles both the change to the global coordinate and placement in the 3D scene.

The user can be further supported in the process of interactively placing virtual objects. By using results from stereo vision, constraints can be imposed on where the user is allowed to specify points representing physical locations in 3D space. Once a point has been specified in one image, the *epipolar constraint* (Shapiro, Zisserman et al. 1995) determines the line in the second image on which the projection of this point must lie. The user specification of the point in the second image can be “snapped” to the nearest point on this epipolar line. Additionally, techniques for constraining points to be collinear or coplanar with physical edges and surfaces are available (Kutulakos and Vallino 1996).

In an augmented view of the scene visual interactions between real and virtual objects must be considered. The affine projection matrix,  $\Pi_{4 \times 4}$ , will correctly handle hidden surface elimination within a virtual object (Figure 4a). It will also cause rendering algorithms to correctly occlude virtual objects that are behind other virtual objects (Figure 4b). Hidden surface removal does not occur when a real object occludes a virtual one (Figure 4c) because there is no information about the geometric relationship between these objects (Wloka and Anderson 1995). If an affine model of a real object is included as another virtual object and rendered in a key color the occlusions are resolved by chroma or luminance keying (Figure 4d). A method for directly creating an affine model for a real object is described in Section 5.

**Desired position** for Figure 4

## **4. The University of Rochester Augmented Reality Systems**

Augmented reality systems based on affine representations have been developed in the Department of Computer Science at the University of Rochester. Conceptually, our augmented reality systems consist of a tracking and graphics subsystems that work together. The major differences between the two systems that have been built to date are in feature tracking and the technology used for viewing the augmented reality image (Azuma 1997).

### **4.1 A monitor-based augmented reality system**

A block diagram of our monitor-based augmented reality system is shown in Figure 5. The augmented reality technique described in this chapter requires the ability to track features in frames throughout a video sequence. It is not dependent on the particular type of feature being tracked. Our affine method does restrict the motion of the features to a rigid motion with respect to each other. Features on multiple objects that are moving relative to each other can not be used. The tracking subsystem provides updates of the affine projection matrix to the graphics system and, as such, can be considered to be an “affine camera position tracking” system. We have implemented trackers that use regions and lines as features (Kutulakos and Vallino 1996). The trackers use Datacube MaxVideo 10 boards as frame grabbers with two standard consumer camcorders (Sony TR CCD-3000) used to view the scene. The trackers are interactively initialized by the user selecting seed points in the regions to be tracked. A search for the boundary of each uniform intensity region proceeds from the seed point using a radially-expanding, coarse-to-fine search algorithm. The detected boundary points are grouped into linear segments using the polyline curve approximation algorithm (Ballard and Brown 1982) and

lines are fitted to the segments using least squares. The region's vertices are located by intersecting the lines fitted to adjacent segments. The actual basis points used to define the affine coordinate system do not directly correspond to visible features. Instead they are the center of mass and the three principal components of the 3D set of vertices of the regions being tracked. The basis points are computed from more than the minimum four points needed to define the affine frame. The inclusion of more than the minimum number of points increases the robustness of the localization of the basis points (Reid and Murray 1996). A Kalman filter (Bar-Shalom and Fortmann 1988) is used for tracking the feature points. The output of the filters estimates the image position and velocity of the projections of the basis points.

**Desired location** for Figure 5

The graphics subsystem is the second component of the system. It is based on a Silicon Graphics workstation with Infinite Reality graphics. The graphics rendering is performed using the OpenGL and Open Inventor graphics libraries. Communication between the tracker and graphics subsystems is via an Ethernet network. The rendered virtual objects are displayed in a graphics window on the workstation console. This image is used as the foreground element of a luminance keying operation (Jack 1993) in a Celect Translator keyer. The background element is the live video signal. The keyer will show live video in all areas of the image where the luminance value of the foreground element is below the key value. This value is set to key on the black background in the graphics image.

System operation is broken into initialization and runtime phases. There are three steps in the initialization phase: (1) graphics-to-tracker alignment, (2) affine basis initialization, and (3) placement of virtual objects. The graphics-to-tracker alignment is necessary due to differences in image coordinates used by the tracker and graphics subsystems. To establish the relationship between these two 2D coordinate systems the correspondence between three points in the two buffers must be known. The graphics system outputs an alignment pattern composed of



three crosses on a black background. This pattern is then merged with the live video signal and digitized by the tracker. (During this alignment sequence the tracker digitizes the merged video signal which is different than normal operation when it works with the original live video.) The user interactively specifies the location of the crosses in the digitized image. From these locations and the known position of the crosses in the graphics image the  $2 \times 3$  *graphics-to-video transform* (Figure 3) is computed. The initialization of the affine coordinate system is performed interactively by the user when the region trackers are initialized as described above. Automatic tracking of these regions by the tracking subsystem commences after initialization. The next step is placement of virtual objects into the scene. This is accomplished interactively using the techniques specified in Section 3.4. With initialization complete the system enters its runtime phase. The tracking subsystem computes the affine projection matrix,  $\Pi_{4 \times 4}$ , and transmits updates to the graphics subsystem at rates between 30 and 60Hz. Using the updated projection matrix the virtual scene is rendered and output to the luminance keyer to create the merged augmented reality image.

The static and dynamic performance of the system was measured. Static performance was measured to determine the misregistration errors caused by the affine approximation to perspective projection and any distortions introduced by the camera lens. The ground truth values were gotten from the image projections of vertices on a physical object. The projections were manually specified in the sequence of images from approximately 50 camera positions. The camera was moved in a roughly circular path around the object at multiple distances ranging to 5m. Camera zoom was used to maintain a constant image size as the distance from the object increased. Four points were selected to define the affine basis through the entire set of images. From these points the affine projection matrix in each image was computed. The affine coordinates of the remaining vertices were calculated using the *affine reconstruction property* from two images in the sequence. The projections of the non-basis points in the other images of

the sequence were then computed and compared to the manually specified points.

Misregistration errors range up to 15 pixels with the larger errors seen for shorter distances to the object. This is as expected with the weak perspective approximation to perspective projection.

Dynamic performance is measured while the system is running and quantifies the misregistration error of overlays caused not only by the factors discussed in the previous paragraph but also by latencies in the system's real-time operation. The test was initialized by calculating the affine coordinates of a feature on a real object from its projection in two images. This feature was not one used for defining the affine coordinate system. The augmented image was a white dot that with perfect registration would align with the feature in the merged view. Two correlation based trackers provided independent measurements of the position of the feature in the live video and the white dot in the graphics image. The former was considered the ground truth and the Euclidean distance in image coordinates between the two positions was measured as the dynamic registration error. The test system was manually translated and rotated in an arbitrary fashion for approximately 90 seconds. The mean absolute registration error in the vertical and horizontal directions was 1.74 and 3.47 pixels, respectively.

## **4.2 Using a video see-through display**

The user's perception of being present in the augmented reality scene can be increased by using either a video or optical see-through display (Azuma 1997). The technique of augmenting reality with affine representations immediately lends itself to operation with a video see-through display. Since neither position information nor camera calibration parameters are needed, creating a video see-through display is simply a matter of attaching cameras to a standard virtual reality head-mounted display (HMD). We created our display by mounting two Panasonic miniature color CCD cameras with 7.5mm lens on a Virtual Research VR4 HMD. To obtain a properly fused stereo view of the real scene the user manually adjusts the fixtures

holding the two cameras to correctly orient them. This is the only additional system initialization step needed. The system block diagram is shown in Figure 6.

**Desired location** for Figure 6

The features being tracked by cameras on the HMD can undergo large scale shifts between video frames due to motion of the user's head (State, Hirota et al. 1996). Because the region-based tracker had difficulty tracking these motions we implemented new trackers based on color blob detection. Color-based tracking also allows for increased flexibility in selecting the real scene because features are simply colored marker dots placed in the scene rather than the high contrast regions used previously. The tracking subsystem is based on a Datacube MV200 image processing system with a Digicolor board for color frame grabbing. The video frames are digitized in the hue-saturation-value (HSV) colorspace where segmentation of color features is performed via table look-up. After performing blob coloring (Ballard and Brown 1982) and centroid calculations the tracker computes the affine projection matrices. These operations are performed at a 30 Hz rate. The registration errors and latencies in this implementation are comparable to those found on our original system using region-based trackers.

The HMD-based system incorporates a new graphics subsystem running on a Silicon Graphics Indigo2 workstation with High Impact graphics and Impact video. The video hardware provides a luminance keyer for merging the graphics on a single channel of live video. An interesting observation made during informal experiments in the laboratory is that the user was unaware of this limitation when the augmented view was presented to the user's dominant eye. Only when asked to close that eye did the user notice the missing augmentation in the non-dominant eye.

## 5. An augmented reality-based interface for interactive modeling

The ability to overlay virtual images on live video leads to an interactive method for building models that can be used as virtual objects. The approach uses the real object as a physical three-dimensional model. Instead of trying to solve the difficult problem of understanding the shape of an object from video images of it, the user uses a hand-held pointer to trace the surfaces of the object for which a model is desired.

First, the global affine coordinate system is computed from the projections of the tracked feature points. No placement of the virtual object is necessary. The user identifies a surface of the object by moving a hand-held pointer in contact with the surface as if painting it (Figure 7a). This motion is tracked from two viewpoints using a normalized correlation technique (Ballard and Brown 1982). Given the projection of the pointer in two images, the *affine reconstruction principle* is applied to compute affine coordinates for that point on the object's surface. This turns the pointer into the equivalent of a 3D digitizer (Foley, van Dam et al. 1990; Tebo, Leopold et al. 1996). Feedback is given to the user by rendering small spheres at the 3D points defined to be on the object's surface and merging this image with the video image (Figure 7b). By looking at this augmented view the user sees areas of the object that have not yet been modeled and those that may require refinement. The augmentation is correctly rendered even if the object is moved (Figure 7c), giving the appearance that the user has applied virtual paint on the surface of the object (Agrawala, Beers et al. 1995). We are currently developing techniques for incremental real-time triangulation, surface representations that can efficiently grow in an incremental fashion, and texture mapping of the object's video image onto the model.

**Desired location** for Figure 7

## 6. Discussion

The technique for augmenting reality using affine representations provides a method for merging virtual objects with live video without magnetic position tracking and camera calibration. While the elimination of these two inputs is an advantage, the technique comes with a set of new requirements and limitations, namely, the requirement for real-time tracking of features, the weak perspective approximation, and the use of a non-Euclidean reference frame.

Augmenting reality with affine representations is limited by the accuracy, response and abilities of the tracking system. To date we have simplified our tracking problem by limiting it to tracking high-contrast objects and easily segmented color blobs. For each video image the tracking subsystem must compute a consistent global coordinate system. This becomes a problem when feature points become occluded or the tracker detects new features in the image. A promising approach to overcome this problem is the Variable State Dimension Filter (VSDF) (McLauchlan and Murray 1995). We are currently testing whether the application of the VSDF allows the tracker to maintain a stable affine coordinate system even when there are a variable number of feature points from one video frame to the next.

This entire system operates by defining all objects in a global affine coordinate system. The technique approximates the camera's perspective projection with a weak perspective model. The validity of this approximation is limited to regions close to the optical axis of the camera, and to objects whose front-to-back distance is small compared to the object's average distance from the camera. The weak perspective assumption can be eliminated if a common projective representation is used instead (Faugeras 1992). A projective representation requires a minimum of five feature points to be tracked instead of the four required by an affine representation. The relative merits of several methods to compute projective structure has been described in the literature (Zisserman and Maybank 1994; Hartley 1995; Rothwell, Csurka et al. 1995; Li, Brady

et al. 1996). The weak perspective approximation also does not account for any distortion in the camera lens. Radial distortion is present to some degree in most lenses. Calculating an appropriate image warp (Mohr, Boufama et al. 1993) or estimating the distortion coefficients in conjunction with the tracking subsystem using a VSDF (McLauchlan and Murray 1996) can compensate for this.

Representing virtual objects in the global affine coordinate system imposes some constraints on system operation. Since the coordinate system in which the virtual objects will be represented is not defined until run-time, the placement of virtual objects can not be done beforehand. No metric information about the real 3D world is used which eliminates the possibility of placing virtual objects based on measurements in the real scene. Instead virtual objects must be placed interactively after the global coordinate system has been determined via methods described in Section 3.4. The graphics system operates in the global affine coordinate system which, in general, is not an orthonormal frame in space. Our system shows that projection computations, z-buffer determination of visible surface and texture mapping can be performed within this affine representation. Other rendering algorithms, such as lighting computations, that require metric information in the form of measurement of angles, can not be performed directly in affine space. Image-based methods can, in principle, provide correct rendering with lighting by linearly combining multiple shaded images of the objects that have been pre-computed (Shashua 1992; Belhumeur and Kriegman 1996; Dorsey, Arvo et al. 1996).

There are also some limitations in our specific implementations that were described in Section 4. The latency in our current color trackers is one video frame for acquisition and one frame for blob coloring and centroid calculation. We have also measured a maximum delay of 90 msec (approximately three video frames) from transmission of a new projection update to rendering of the virtual objects. This results in a total latency on the order of 5 frames. We are investigating methods to perform tracking at 60Hz to reduce the latency introduced by the

tracking subsystem. We are also experimenting with predictive estimation of feature positions (Azuma and Bishop 1994) for mitigating the other latency in our system. Finally, our graphics subsystem only has the hardware capability to perform a single merging operation. For a monitor-based augmented reality system this is not a major limitation but in the HMD system it eliminates the possibility of the user viewing stereo virtual objects. The purchase of additional hardware to provide luminance keying on two video channels will overcome this limitation.

## **7. Conclusions and future work**

The primary challenge for an augmented reality system is to determine the proper rendering and registration of the virtual objects that will be merged with the view of the real scene. This requires computation of the relationships between multiple coordinate systems. Most augmented reality systems use methods that transform these coordinate systems into a common 3D Euclidean frame relying on position sensing, camera calibration and knowing the 3D locations of fiducial markers in the scene. This chapter has presented an alternative method for rendering virtual objects that are registered with a live video image to create augmented reality images. The problem has been reduced to:

- real-time tracking of a set of four or more points that define a global affine coordinate system,
- representing the virtual objects in this coordinate system,
- computing the projections of virtual points in each video image as linear combinations of the projections of the affine basis points.

We are continuing to work in this area to overcome limitations present in the prototype. Affine representations are particularly well suited when apriori knowledge of the environment is not available. To work in this general setting, a tracking subsystem capable of tracking features

in a natural setting is needed. The tracking subsystem should also handle temporary occlusions or permanent disappearance of features and the appearance of new features in images. We are investigating recursive estimation techniques that will compute a consistent global coordinate system while being robust to these perturbations in the feature set. The weak perspective approximation of the camera's perspective projection generates errors in rendering and registration when the system operates outside the range in which the approximation is valid. Representing objects in a common projective frame will remove the limitations of that approximation.

## **8. Acknowledgments**

The authors would like to thank Chris Brown for many helpful discussions and for his constant encouragement and support throughout the course of this work. The financial support of the National Science Foundation under Grant No. CDA-9503996, of the University of Maryland under Subcontract No. Z840902 and of Honeywell under Research Contract No. 304931455 is also gratefully acknowledged.

## **9. References**

Agrawala, M., A. C. Beers and M. Levoy (1995), "3d painting on scanned surfaces," In *Proc. Symposium on Interactive 3D Graphics*, pages 145-150.

Ahlers, K., D. Breen, C. Crampton, E. Rose, M. Tuceryan, R. Whitaker and D. Greer (1994), "An Augmented Vision System for Industrial Applications," Technical Report ECRC-94-39, European Computer Industry Research Center (ECRC) 1994.



- Azuma, R. and G. Bishop (1994), "Improving static and dynamic registration in an optical see-through hmd," In *Proceedings SIGGRAPH '94*, pages 197-204.
- Azuma, R. T. (1997), "A Survey of Augmented Reality," *Presence*, 6(4):355-385.
- Bajura, M. and U. Neumann (1995), "Dynamic Registration Correction in Video-Based Augmented Reality Systems," *IEEE Computer Graphics and Applications*, 15(5):52-60.
- Ballard, D. H. and C. M. Brown (1982), *Computer Vision*, Englewood Cliffs: Prentice-Hall, Inc.
- Barrett, E. B., M. H. Brill, N. N. Haag and P. M. Payton (1992), "Invariant Linear Methods in Photogrammetry and Model-Matching," In J. L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, 277-292. Cambridge, MA: The MIT Press.
- Bar-Shalom, Y. and T. E. Fortmann (1988), *Tracking and Data Association*: Academic Press.
- Belhumeur, P. N. and D. J. Kriegman (1996), "What is the set of images of an object under all possible lighting conditions," In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 270-277.
- Boufama, B., D. Weinshall and M. Werman (1994), "Shape from motion algorithms: a comparative analysis of scaled orthography and perspective," In *Proceedings of the European Conference on Computer Vision*, pages 199-204.
- Byrson, S. (1992), "Measurement and calibration of static distortion of position data from 3D trackers," In *Proceedings SPIE Vol. 1669: Stereoscopic Displays and Applications III*, pages 244-255.
- Dorsey, J., J. Arvo and D. Greenberg (1996), "Interactive design of complex time dependent lighting," *IEEE Computer Graphics and Applications*, 15(2):26-36.
- Faugeras, O. D. (1992), "What can be seen in three dimensions with an uncalibrated stereo rig?," In *Proceedings of Second European Conference on Computer Vision*, pages 563-578.

- Feiner, S., B. MacIntyre and D. Seligmann (1993), "Knowledge-Based Augmented Reality," *Communications of the ACM*, 36(7):53-62.
- Foley, J. D., A. van Dam, S. K. Feiner and J. F. Hughes (1990), *Computer Graphics Principles and Practice*, Reading, MA: Addison-Wesley Publishing Co.
- Hartley, R. I. (1995), "In Defence of the 8-point Algorithm," In *Proceedings 1995 IEEE International Conference on Computer Vision*, pages 1064-1070.
- Hoff, W. A., K. Nguyen and T. Lyon (1996), "Computer Vision-Based Registration Techniques for Augmented Reality," In *Proceedings SPIE Vol. 2904: Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*, pages 538-548.
- Jack, K. (1993), *Video Demystified: A Handbook for the Digital Engineer*, Solana Beach, CA: HighText Publications Inc.
- Janin, A. L., D. W. Mizell and T. P. Caudell (1993), "Calibration of head-mounted displays for augmented reality applications," In *Proceedings IEEE Virtual Reality Annual International Symposium '93*, pages 246-255.
- Koenderink, J. J. and A. J. van Doorn (1991), "Affine Structure from Motion," *Journal of the Optical Society of America A*, 8(2):377-385.
- Kutulakos, K. N. and J. R. Vallino (1996), "Affine Object Representations for Calibration-Free Augmented Reality," In *Proceedings of 1996 IEEE Virtual Reality Annual International Symposium*, pages 25-36.
- Li, F., M. Brady and C. Wiles (1996), "Fast Computation of the Fundamental Matrix for an Active Stereo Vision System," In *Proceedings of the Fourth European Conference on Computer Vision*, pages 157-166.

McLauchlan, P. F. and D. W. Murray (1995), "A unifying framework for structure and motion recovery from image sequences," In *Proceedings of the 5th IEEE International Conference on Computer Vision*, pages 314-320.

McLauchlan, P. F. and D. W. Murray (1996), "Active Camera Calibration for a Head-Eye Platform Using the Variable State-Dimension Filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):15-21.

Mellor, J. P. (1995a), *Enhanced Reality Visualization in a Surgical Environment*, Masters thesis, AI Lab, Massachusetts Institute of Technology.

Mellor, J. P. (1995b), "Realtime Camera Calibration for Enhanced Reality," In *Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine '95 (CVRMed '95)*, pages 471-475.

Mohr, R., B. Boufama and P. Brand (1993), "Accurate projective reconstruction," In J. L. Mundy, A. Zisserman and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, 257-276: Springer-Verlag.

Mundy, J. L. and A. Zisserman (1992), *Geometric Invariance in Computer Vision*, Cambridge, MA: The MIT Press.

Neumann, U. and Y. Cho (1996), "A Self-Tracking Augmented Reality System," In *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, pages 109-115.

Reid, I. D. and D. W. Murray (1996), "Active Tracking of Foveated Feature Clusters Using Affine Structure," *International Journal of Computer Vision*, 18(1):41-60.

Rothwell, C., G. Csurka and O. Faugeras (1995), "A Comparison of Projective Reconstruction Methods for Pairs of Views," In *Proceedings 1995 IEEE International Conference on Computer Vision*, pages 932-937.

Shapiro, L., A. Zisserman and M. Brady (1995), "3D Motion Recovery via Affine Epipolar Geometry," *International Journal of Computer Vision*, 16(2):147-182.

Shashua, A. (1992), *Geometry and Photometry in 3D Visual Recognition*, PhD thesis, MIT.

Shashua, A. (1993), "Projective Depth: A Geometric Invariant for 3D Reconstruction from Two Perspective/Orthographic Views and For Visual Recognition," In *Proceedings 1993 IEEE International Conference on Computer Vision*, pages 583-590.

State, A., D. T. Chen, C. Tector, A. Brandt, H. Chen, R. Ohbuchi, M. Bajura and H. Fuchs (1994), "Case Study: Observing a Volume Rendered Fetus within a Pregnant Patient," In *Proceedings of the 1994 IEEE Visualization Conference*, pages 364-368.

State, A., G. Hirota, D. T. Chen, W. F. Garrett and M. A. Livingston (1996), "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 429-438.

Tebo, S. A., D. A. Leopold, D. M. Long, S. J. Zinreich and D. W. Kenedy (1996), "An optical 3D digitizer for frameless stereotactic surgery," *IEEE Computer Graphics and Applications*, 16(1):55-64.

Thompson, W. B. and J. L. Mundy (1987), "Three-dimensional model matching from an unconstrained viewpoint," In *Proceedings of the IEEE 1987 Robotics and Automation Conference*, pages 208-220.

Tsai, R. Y. (1987), "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Transactions of Robotics and Automation*, RA-3(4):323-344.

Tuceryan, M., D. S. Greer, R. T. Whitaker, D. E. Breen, C. Crampton, E. Rose and K. H. Ahlers (1995), "Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System," *IEEE Transactions on Visualization and Computer Graphics*, 1(3):255-273.

Uenohara, M. and T. Kanade (1995), "Vision-Based Object Registration for Real-Time Image Overlay," In N. Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medicine: CVRMed '95*, 14-22. Berlin: Springer-Verlag.

Ullman, S. and R. Basri (1991), "Recognition by Linear Combinations of Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992-1006.

Weinshall, D. and C. Tomasi (1993), "Linear and Incremental Acquisition of Invariant Shape Models from Image Sequences," In *Proceedings 4th IEEE International Conference on Computer Vision*, pages 675-682.

Wiles, C. and M. Brady (1996), "On the Appropriateness of Camera Models," In *Proceedings of the Fourth European Conference on Computer Vision*, pages 228-237.

Wloka, M. M. and B. G. Anderson (1995), "Resolving Occlusion in Augmented Reality," In *Proceedings 1995 Symposium on Interactive 3D Graphics*, pages 5-12.

Zisserman, A. and S. J. Maybank (1994), "A Case Against Epipolar Geometry," In J. L. Mundy, A. Zisserman and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, 69-88. Berlin, Germany: Springer-Verlag.

## Figure captions

Figure 1 - Components in an Augmented Reality System. (a) The multiple coordinate systems that must be registered are shown. Several types of augmented reality systems exist (Chapter ??? Azuma). This diagram depicts a system using a monitor-based display or video see-through head-mounted display. (b) View of an affine wireframe model correctly overlaid on a small box. (c) Example of a misregistration of a virtual object with the real scene. The virtual wireframe is not correctly registered on the small box.

Figure 2 - Properties of affine point representations. The tracked features  $p_0, p_1, p_2, p_3$  define an affine coordinate frame within which all world points can be represented: Point  $p_0$  is the origin, and points  $p_1, p_2, p_3$  are the basis points. The affine coordinates of a fifth point,  $p$ , are computed from its projection in (a) and (b) using Property 2.  $p$ 's projection in (c) can then be computed from the projections of the four basis points using Property 1.

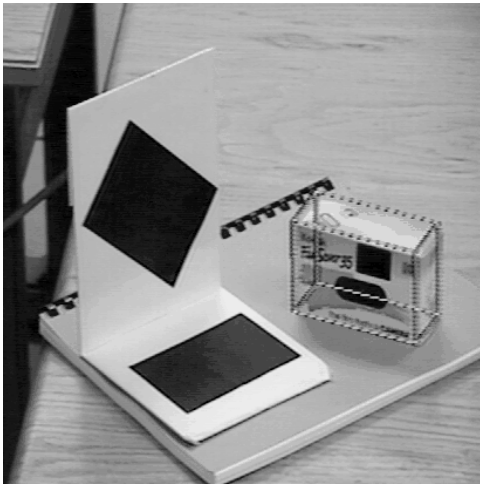
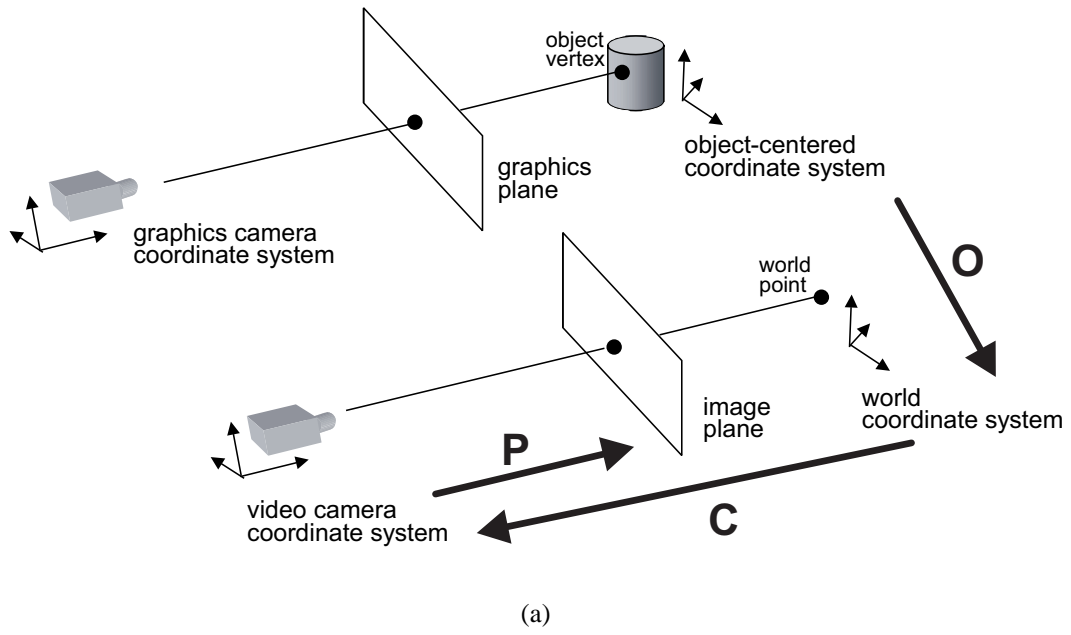
Figure 3 - Procedure for rendering virtual objects showing the coordinate systems and transforms involved.

Figure 4 – Visible-surface rendering of affine virtual objects. The virtual towers were represented in OpenInventor™. Affine basis points were defined by the centers of the circular markers. The virtual towers were defined with respect to those points. (a) Initial augmented view. (b) Augmented view after a clockwise rotation of the object containing the affine basis points. (c) Hidden-surface elimination occurs only between virtual objects; correct occlusion resolution between physical and virtual objects requires information about the geometric relations between them (Wloka and Anderson 1995). (d) Real-time visible surface rendering with occlusion resolution between virtual and real objects. Visibility interactions between the virtual towers and the L-shaped object were resolved by first constructing an affine graphical model for the object. By painting the entire model a fixed background color and treating it as an additional virtual object, occlusions between that object and all other virtual objects are resolved via chroma- or luminance-keying. Affine models of real objects can be constructed using the interactive modeling technique of Section 5.

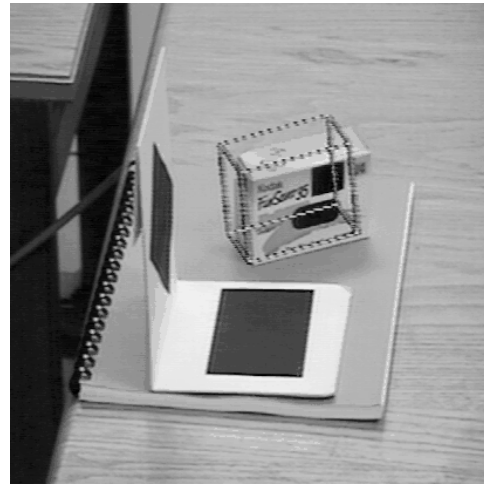
Figure 5 - Configuration of a monitor-based augmented reality system

Figure 6 - Configuration using a video see-through display. The affine basis in the example images was defined by the four circular markers, which were tracked in real time using color segmentation. The markers were manually attached to objects in the environment and their 3D configuration was unknown.

Figure 7 - Interactive 3D affine modeling. Live video is provided by two camcorders whose position and intrinsic parameters were neither known in advance nor estimated. (a) An easily-distinguishable hand-held pointer is moved over the surface of an industrial part. The dark polygonal regions are tracked to establish the affine basis frame. The regions were only used to simplify tracking and their Euclidean world coordinates were unknown. (b) Visualizing the progress of 3D stenciling. The augmented display shows the user drawing a virtual curve on the object's surface in real-time. (c) When the object is manually rotated in front of the two cameras, the reconstructed points appear "locked" on the object's surface, as though the curve traced by the pointer was actually drawn on the object.



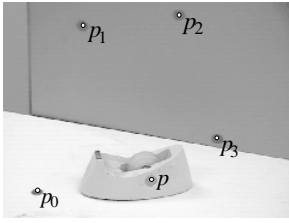
(b)



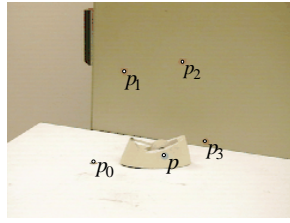
(c)

Figure 1

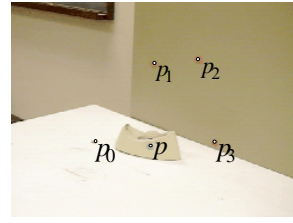




(a)



(b)



(c)

Figure 2

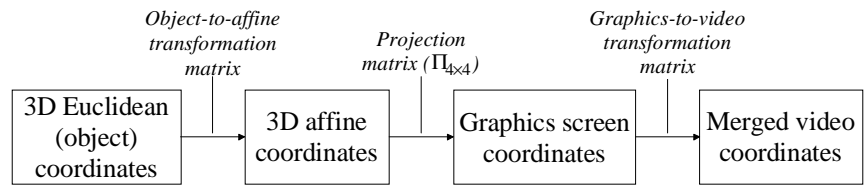
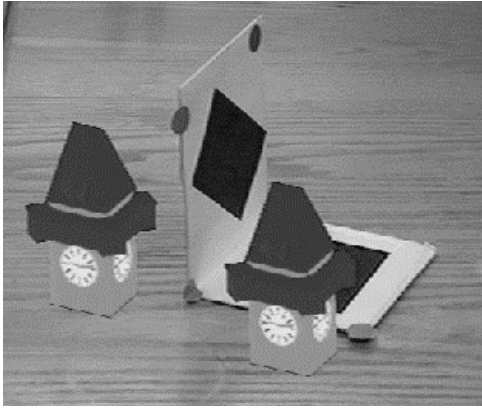
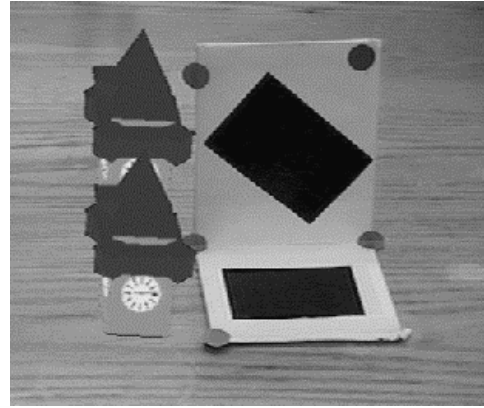


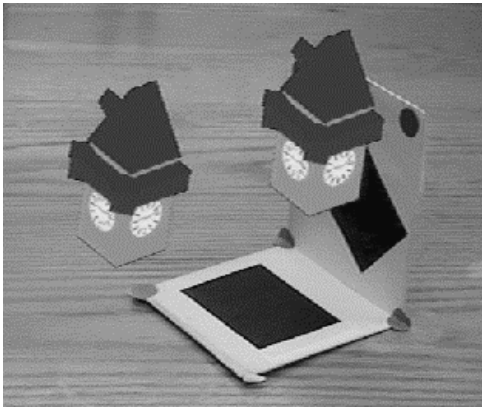
Figure 3



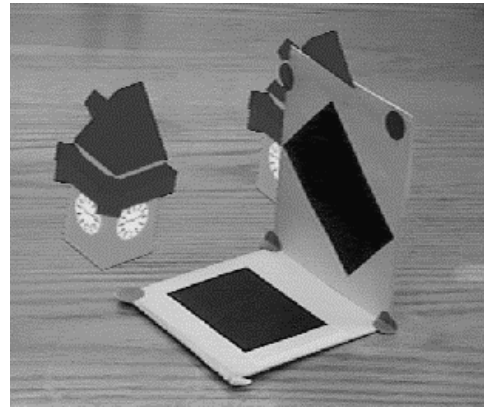
(a)



(b)



(c)



(d)

Figure 4

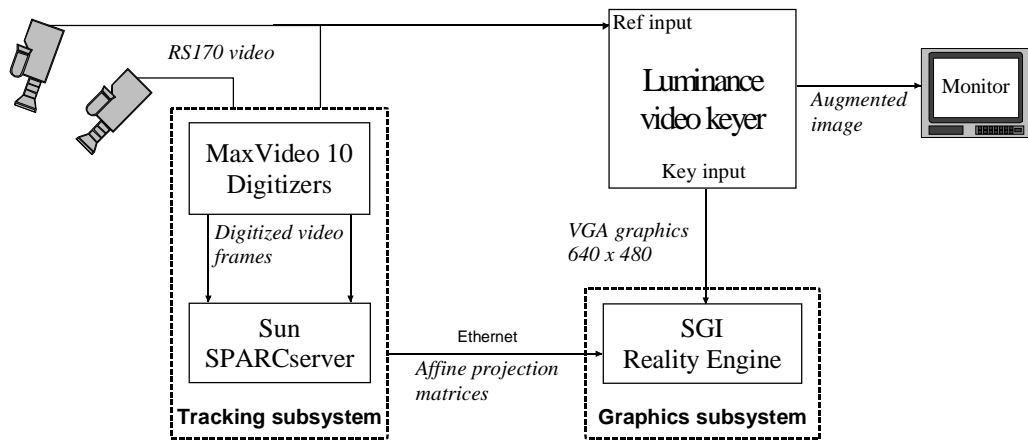


Figure 5

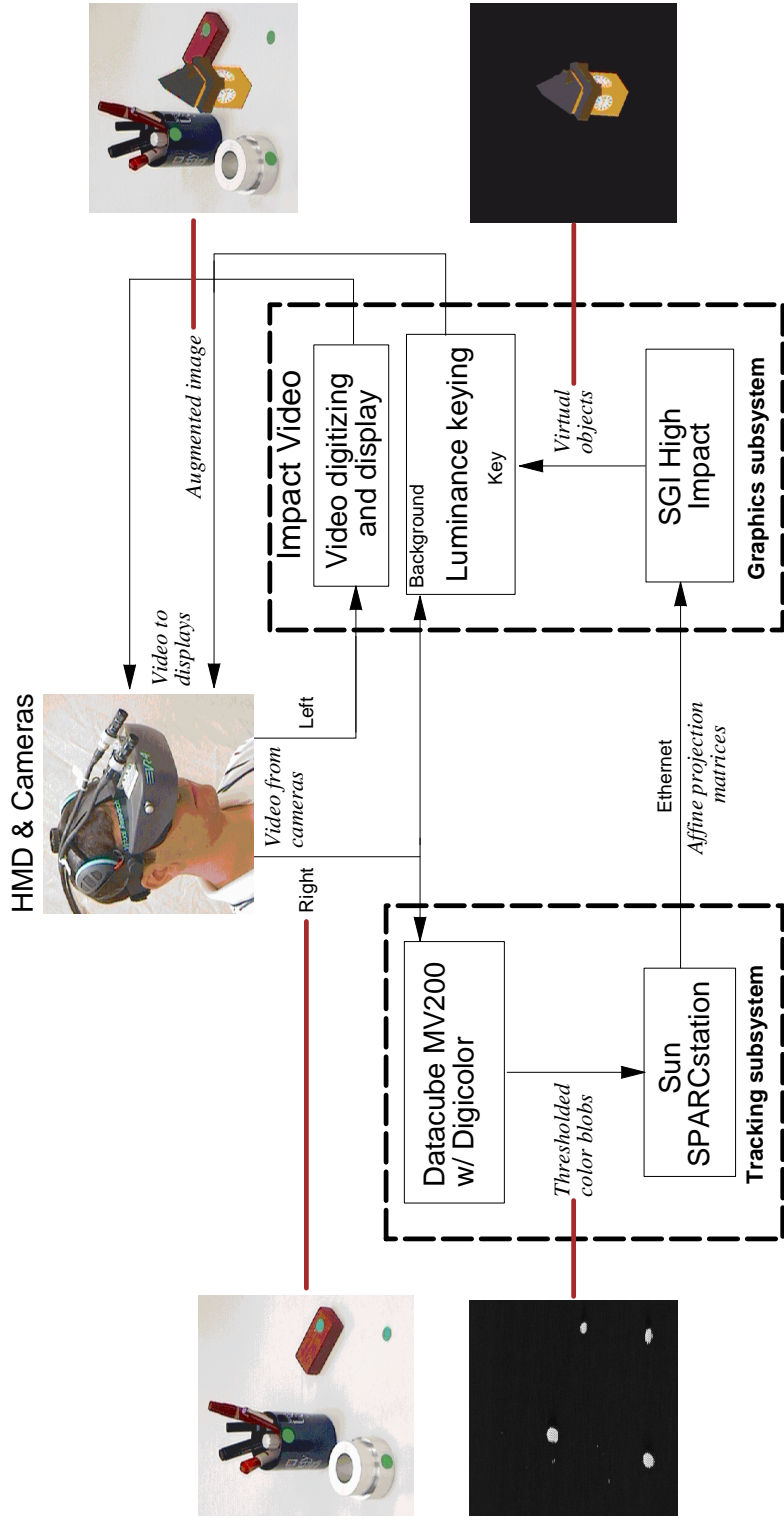
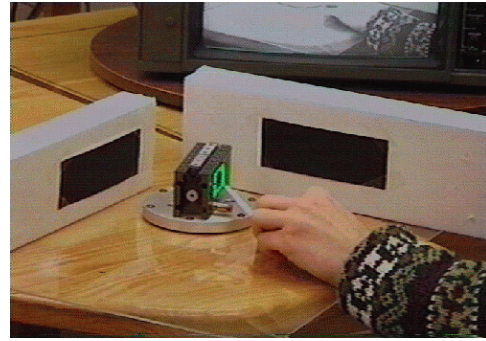


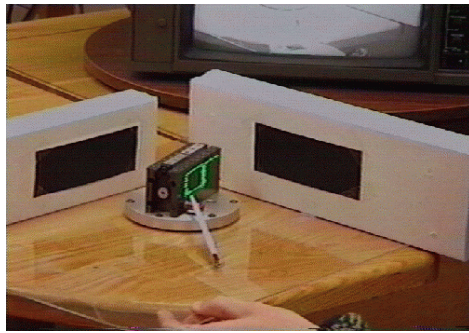
Figure 6



(a)



(b)



(c)

Figure 7