

9 Adapting to Registration Error in an Intent-Based Augmentation System

Cindy Robertson, Blair MacIntyre

Augmented Environments Lab, College of Computing and GVU Centre, Georgia Institute of Technology, Atlanta, GA, 30332, USA

An ongoing research problem in Augmented Reality (AR) is to improve tracking and display technology in order to minimize registration errors. However, registration is not always necessary for users to understand the intent of an augmentation, especially in industrial applications where the user and the system have extensive semantic knowledge of the environment. In this chapter, we review the ideas of communicative intent developed for desktop graphical explanation systems by Seligmann and Feiner, and discuss how these approaches are the basis for our hypothesis that semantic knowledge of a scene can be used to ameliorate the effects of registration errors. We describe a set of AR visualization techniques for augmentations that adapt to changing registration errors. We first define a set of strategies that use semantic knowledge of the augmentation to enhance the augmentations with additional contextual cues. These context cues help users understand the intent of the augmentation in the presence of registration error. We then introduce algorithms that use features and feature points on objects to control these strategies in the presence of changing registration errors. Finally, these algorithms and techniques are demonstrated in four maintenance situations that challenge a user's ability to interpret the semantics of a scene.

Keywords: *Augmented Reality; Registration Error; Communicative Intent; Human Computer Interaction; Visualization*

9.1 Introduction

Numerous researchers have demonstrated the potential of AR (the use of see-through head-worn displays to overlay graphics on the physical world) as a powerful user-interface paradigm for manufacturing applications. The key advantage of these systems is that they situate the graphics *in situ* and support

hands-free interaction. Tang et al. found that overlaying 3D instructions in an assembly task reduced the error rate for that task by 82%, particularly diminishing cumulative errors, as compared to printed instructions, instructions on an LCD monitor, and instructions on a see-through head-mounted display (HMD) (Tang et al. 2003). They also found that mental effort decreased for the AR condition. However, one significant hurdle that must be crossed when creating AR applications is to register the graphics with objects in the world: to integrate graphics accurately with physical objects, both the user and the objects must be accurately tracked (at least with respect to each other), and the whole system (including the HMD) must be accurately calibrated. In some domains, such as medicine, accurate registration is required. However, we believe that in many situations, such as those that exist in industrial applications, precise registration is not as critical. For example, if a maintenance system, such as the KARMA system (Feiner et al. 1993), instructs a repair person to move a lever on a machine, and there is only one such lever in the area being augmented (such as a lever to open the top of a printer), precise registration may not be necessary.

In other words, registration requirements are not absolute; they depend on the domain, the specific context of use, and the communicative intent of the augmentation. Seligmann and Feiner described communicative intent as “the audience interpretation and consequent actions that the communicator wishes to elicit and are comprised of a set of communicative goals” (Seligmann and Feiner 1991). Specifically, the communicative intent “provides a high-level description of what is to be communicated,” and the communicative goals “drive the process that determines what information (set of objects and properties) to use to satisfy intent.”

Seligmann and Feiner explored communicative intent deeply in the context of the automated design of graphical presentations with the IBIS system (Seligmann 1993), and adapted IBIS to support the interactive AR system KARMA (Feiner et al. 1993). In KARMA, they took into account the basic differences between AR and static 3D images: user control of the camera and the presence of unchangeable physical objects. However, they designed their system assuming perfect tracking and registration, ignoring the inevitable uncertainty in the system created by imperfect sensors and displays. We believe that in many cases, it should be possible to use knowledge of the uncertainty of the physical world to design augmentations that a user can understand, even when the registration error arising from this uncertainty is significant.

Our work is based on the observation that humans are good at leveraging contextual cues to interpret ambiguous situations. Consider the simple KARMA example mentioned above. In such a maintenance system, if the augmentation of the lever is not registered with the physical lever, but there is only one lever and the augmentation is near it, a repair person should understand that the augmentation refers to the physical lever. Conversely, if there are multiple levers, but the one in question is below a unique feature (such as a large button), adding a representation of the button to the augmentation may be enough to allow the human to choose the correct lever.

9.1.1 The AIBAS Project

The goal of AIBAS (an *Adaptive Intent-Based Augmentation System*) is to understand how the semantic knowledge of a scene can be leveraged to simplify the creation of AR applications that work well in real-world situations with “good enough” tracking and registration. In this chapter, we demonstrate how such knowledge can be used to reduce the impact of registration errors by supporting the programmer in creating augmentations that contain sufficient visual context for a user to understand the intent of the augmentation. Our goal is to empower programmers by providing a conceptual framework, and the associated tools, to support the creation of augmentations that function in the presence of registration error. Our longer-term goal is to create a toolkit with “AR widgets” that allow a wide variety of AR applications to be built in a straightforward manner. In addition, while we have no current plans to implement a knowledge-based illustration system such as IBIS (the system on which KARMA was based), these techniques could be easily integrated into such a system (refer to section 9.8 for a discussion of our future directions for AIBAS).

We assume that the underlying system provides the programmer with a continuous estimate of registration error, using the techniques we have described in our previous work (MacIntyre et al. 2002). We also assume the application has knowledge of the domain (such as models of the important physical objects) and the communicative intent of the augmentations; these techniques are not intended to operate on arbitrary graphics in the absence of semantic knowledge.

The contribution of this chapter is twofold. First, we discuss the concept of *communicative intent* as it relates to our belief that the semantic knowledge of a scene can be used to ameliorate registration error, an important problem faced by AR systems. In particular, Seligmann and Feiner used a collection of *style strategies* to implement the communicative goals; we describe how each of them can be adapted to provide this context in the presence of registration error. While the specific techniques discussed later in the chapter do not rely directly on Seligmann and Feiner’s framework, it represents a clear way to think about this problem, and a system built using it would contain exactly the right level of semantic information.

Second, we discuss our approach, namely to add visual context to an augmentation to help the user understand the intent of the augmentation when it is not perfectly registered. Our approach relies on the system having sufficient knowledge of the intent of the augmentation, and on the viewer’s ability to make sense of ambiguous situations when provided with sufficient context. We present four examples showing how these strategies would be used in practice, and describe the algorithms we used to implement them. The examples cover a range of common AR situations and require different kinds of context.

In the next two sections, we discuss related work, and review the idea of communicative intent (and communicative goals and style strategies in particular) introduced by Seligmann and Feiner. We give examples of AR systems that have (implicitly or explicitly) used each of these goals and strategies, and discuss the impact of registration error on the style strategies. We then describe our strategies

for adding context to augmentations, and briefly discuss the architecture and implementation of our system. Finally, we present four detailed examples, and close with conclusions and a discussion of future work.

9.2 Related Work

There has been significant work done on reducing registration error in AR systems (Azuma and Bishop 1994; Holloway 1997; Hoff 1998), far too much to summarize here. Analyzing and estimating error bounds, especially in vision-based tracking systems, is not a new idea. Holloway's work is perhaps the best-known analysis of registration error; we use his terminology and framework. Hoff used error estimates as the basis for fusing multiple sensors; like him, we represent error estimates as probability distributions. Our work is complementary to research aimed at reducing registration error; as long as there is registration error, the techniques described here will be useful.

AIBAS builds on our previous work on adapting to registration errors (MacIntyre et al. 2002). In that system, we used estimates of the transformation errors in an AR system (such as those introduced by tracker measurement errors, as well as errors arising from measurement and calibration error) to estimate the registration error of points in a 3D world on a 2D display. We showed how to perform simple modifications to interesting regions of the display in an AR system, such as expanding and contracting the 2D convex hull of the projection of a virtual object to find the area of the screen the object *might* and *should* occupy, respectively. In this chapter, we expand on the idea of using registration error estimates. In AIBAS, we use these estimates to guide the creation of intent-based augmentations, rather than focusing on simple graphical transformations as we have previously done.

Other researchers are also concerned with creating AR systems that work in the presence of imperfect tracking and registration errors. A common approach is to build the AR system assuming a worst-case error, especially in mobile AR systems that use GPS for tracking. For example, the Touring Machine uses textual labels as augmentations, which do not need to be accurately registered (Feiner et al. 1997). Julier and his colleagues tried to make their AR interface more understandable by minimizing the amount of unnecessary information presented to the user using a region-based information-filtering algorithm (Julier et al. 2000). Their system dynamically responds to changes in the environment and the user's state, but does not explicitly take registration error into account. Andre and Cutler (1998) used rings to represent uncertainty in the location of an object: the size of the ring was dependent upon the level of uncertainty. This approach works well for individual objects with small amounts of error, but we wish to handle varying amounts of error with multiple objects.

AIBAS was inspired by the IBIS and KARMA systems. IBIS is a knowledge-based system that generates graphics to explain the communicative intent of a scene. KARMA, an AR system based on IBIS, generates AR illustrations for

maintenance tasks on a computer printer. Neither IBIS nor KARMA takes registration error into account. AIBAS is designed to support the graphics generation component of a system like KARMA, allowing its illustrations to dynamically adapt to changing registration errors.

Some motivations for our approach come from the field of Gestalt psychology, which seeks to explain how humans perceive objects in terms of perceptual outcomes, rather than focusing on cognitive mechanisms (Maren and Ali 1988). It defines different types of perceptual relationships that play a role in how a viewer structures the elements in a scene: enclosure of one region by another, good continuation (or co-linearity), proximity, and similarity (of size, shape, intensity, hue, orientation and other features.) These concepts are formed by the observation that people inherently make associations and references based on the scene as a whole. Therefore, we believe that if an AR system generates a well-structured collection of perceptual cues, viewers will perceive the intent correctly.

Table 9.1. Communicative goals

Communicative Goal	Description of Goal	Examples from Existing AR Systems
Show	Introduce a new object to the user or to familiarize the user with an object. Requires that the representation of the object be visible and recognizable	Transparent surfaces, selective rendering (Darken and Cevik 1999; Dinsmore et al. 1997) Transparent surfaces, depth info used to determine whether virtual object is occluded by real object (Furmanski et al. 2002; Kanbara et al. 2000) Inset windows (Billinghurst et al. 1998; Darken and Cevik 1999; Johnson et al. 1999) User defined/ interactive viewpoints (Darken and Cevik 1999; Dinsmore et al. 1997; Moezzi et al. 1996; Risch et al. 1996) Cutaway view (Bajura et al. 1992; Furmanski et al. 2002; State et al. 1996) Superimposition with anatomy (Argotti et al. 2001) Superimposed X-ray images on body parts (Navab et al. 1999a)
Property	Show the specific properties of an object that may be used to describe it	Displays different brain material in different colours (Grimson et al. 1996)
State	Depict features of an object that show it is in a specific state	Uses arrows to show velocity data (Ogi and Hirose 1996) Uses streamlines, isosurfaces, and cutting planes to show velocity vectors and density scalars around an aircraft carrier (Bryson et al. 1997) Annotations vary based on a person's distance from the user (Newman et al. 2001)
Location	Show where an object is. Usually implies that the object be shown in a particular context so the user may better interpret the object's location	Inset window (Billinghurst et al. 1998; Darken and Cevik 1999; Johnson et al. 1999) Shows where certain parts of the brain are located inside the head (Grimson et al. 1996) Shows internal anatomy superimposed on body (Argotti et al. 2001; Navab et al. 1999a) Shows internal structures of a building contained in industrial drawings (Navab et al. 1999b)

Table 9.1. (cont.)

Communicative Goal	Description of Goal	Examples from Existing AR Systems
Reference	Provide additional objects in the scene in order to provide a cohesive view	Labels objects with labels that do not interfere with other objects (Bell et al. 2001)
Change	Show how an object changes state and the differences between these states	Uses dotted lines to show suggested shots in a pool game (Starner et al. 1998)
Relative-location	Show a group of objects in a common context, and the relationship between these objects	Displays different parts of the brain inside the head, using colour to distinguish them (Grimson et al. 1996) Displays where internal structures are located inside of a body (Argotti et al. 2001; Bajura et al. 1992; Dinsmore et al. 1997; Navab et al. 1999a) Displays where internal structures of a building are located inside factory (Navab et al. 1999b) Uses transparent walls to resolve depth ambiguity (Holloway 1997)
Identify	Help the user figure out what they are looking at by positioning identifier objects over or near object	Labels objects (Bell et al. 2001; Newman et al. 2001; Simon and Berger 2002) Lines to show linkage (Risch et al. 1996)
Action	Show the user how to perform an action in the real world that enables the object in the virtual world to reach a certain state	Uses arrows to show movement (Feiner et al. 1993; Reiners et al. 1998) Aids navigation with virtual signposts, a 2D map, compass arrows, or turning signals (Navab et al. 1999a) Uses virtual cylinder to guide a needle to the correct path (Sauer et al. 2000; Vogt et al. 2002)
Move	Show how an object is to be manipulated	Uses arrows to show movement (Feiner et al. 1993) Shows the steps in placing a virtual cube on a real object (Kutulakos and Vallino 1996)
Enhancement	Show additional objects in the scene that may not have any direct effect on the action taking place in the scene	Many systems add additional virtual objects.

Piaget and Inhelder (1956) found that children first learned to recognize surfaces and their outlines. Several years later, Rock (1975) and Zusne (1970) found that the silhouette of an object is the determining factor for the recognition of the object. Gooch and Willemsen (2002) found that humans could perceive depth at 66% of the intended distance in an immersive environment that renders objects with only feature edges. Woods (1995) found that graphical representations such as shape, symbols, size, colour and position are effective in information visualization because they are mentally economical. Interrante et al. and Pizer (1995) found that adding sparse opaque textures to transparent surfaces can help make its location in space more explicit. They also found that ridge and

valley lines carry geometrical and perceptually relevant information. This psychology and visualization research has provided us with inspiration for the actual design of the visualizations.

9.3 Communicative Intent in AR

Communicative intent is the conceptual foundation of this research; some high-level notion of the semantics of each augmentation is necessary if the system is to adjust the graphical display to account for the current viewing context. As originally defined, intent is specified as a collection of goals that an augmentation is trying to accomplish. Style strategies describe the visual effects used to achieve each of these communicative goals.

Table 9.2. Style Strategies

Style Strategy	Description of Style Strategy	Examples from Existing AR Prototypes	Strategies in the Presence of Registration Errors
Include	Used to represent objects. If the object has subparts, they are typically included.	Includes augmentations of new pipes to allow for possible collision detection between existing structures (Navab et al. 1999b)	Highlighting physical objects near the included virtual objects will reinforce their intended relationship with the physical world.
Visible	Used to ensure the visibility of an object. Use cutaway view to show hidden objects when “camera” is under user control (e.g., in head-tracked AR).	Transparent surfaces, selective rendering (Darken and Cevik 1999; Dinsmore et al. 1997; Newman et al. 2001) Depth info used to determine whether virtual object is occluded by real object, use transparent surfaces (Furmanski et al. 2002; Kanbara et al. 2000) Inset window (Billinghurst et al. 1998; Darken and Cevik 1999; Johnson et al. 1999) User defined/interactive viewpoints (Darken and Cevik 1999; Dinsmore et al. 1997; Moezzi et al. 1996; Risch et al. 1996) Cutaway view (Bajura et al. 1992; Furmanski et al. 2002; State et al. 1996) Shows internal structure of leg (Argotti et al. 2001) Shows internal structure of a body via X-ray images (Navab et al. 1999a)	Highlighting physical objects near the cutaway will reinforce the relationship between the cutaway (and thus the exposed object) and the physical world.

Table 9.2. (cont.)

Style Strategy	Description of Style Strategy	Examples from Existing AR Prototypes	Strategies in the Presence of Registration Errors
Find	Used to help the user locate a certain object.	Arrow point toward off-screen landmarks (Feiner et al. 1997) Annotations to locate a person or object (Newman et al. 2001)	Target of find may or may not be visible due to registration error. Additional context or more general directions may disambiguate.
Label	Used to help identify objects.	Labels to point out objects that do not interfere with other objects (Bell et al. 2001; Newman et al. 2001; Simon and Berger 2002) Lines to show linkage (Risch et al. 1996)	Labels should avoid obscuring all possible locations of objects (Bell et al. 2001). Target of lines may be ambiguous, so either point to unambiguous part of object, or highlight region object might occupy.
Recognizable	Used to show certain properties of an object so it is recognizable.	Displays different brain material in different colours (Grimson et al. 1996)	Highlight sufficient physical context to make identity of recognizable objects clear
Focus	Used to draw the user's attention to the object.	Highlights and labels have been used, as in other strategies.	Highlight sufficient physical context to make identity of focal objects clear
Subdue	Used to show object is not currently significant.	No examples of use. In video-mixed AR, could desaturate or blur object	Be careful not to subdue significant objects in video due to error
Visual Property	Used to render the object such that certain property values are shown.	Size of annotations denote a person's location from a user (Newman et al. 2001)	Analogous issues to Include and Subdue.
Ghost	Used to show an object in a scene without fully occluding other objects.	Uses ghosting to show motions of different objects (Feiner et al. 1993; Sung et al. 1999) Ghosted walls resolve depth ambiguity (Holloway 1997)	Analogous issues to Include.
Highlight	Used to attract the user's attention to an object using distinctive visual cues.	Highlights icons resulting from a search (Risch et al. 1996) Overlaid graphics in bright, distinguishable colours (Feiner et al. 1993, 1995; Grimson et al. 1996; Klinker et al. 2001)	In the face of moderate error, highlights can be expanded to encompass the possible location of an object. (MacIntyre et al. 2002)

Table 9.1 lists and defines the communicative goals that we are currently studying, most of which were borrowed from (Seligmann 1993). These goals have been shown to be applicable in AR domains. We provide examples of augmentations from existing AR systems for each of the goals listed (even though most of the system authors did not explicitly use the framework of communicative intent when designing their augmentations).

Each example augmentation uses one or more style strategies to achieve these goals. Table 9.2 lists and defines a collection of possible style strategies, also borrowed from Seligmann, that are used in this research. The table also gives examples of how these strategies have been used in AR prototypes, and discusses the implications of registration error for each strategy. Notice that style strategies do not define the graphical techniques to be used, but instead specify a strategy for achieving a goal. This means that a similar collection of strategies may be used for purely virtual 3D images and for AR, even though the techniques for implementing the strategies may be different. For example, to *highlight* an object in a 3D scene, we might render it in a bright, unnatural colour, or cause it to flash between multiple colours. However, in an AR system, the physical object is present, so we should not render it, as doing so may obscure the physical world. Rather, we might render its edges or silhouette in a bold colour to allow most of the physical object to be seen.

Table 9.2. (cont.)

Style Strategy	Description of Style Strategy	Examples from Existing AR Prototypes	Strategies in the Presence of Registration Errors
Context	Used to include other context objects in a scene or to generate at least one illustration-object that corresponds to an ancestor of the object and all its subparts.	Inset window (Billinghurst et al. 1998; Darken and Cevik 1999; Johnson et al. 1999) Shows interior of the brain relative to the skull (Grimson et al. 1996) Overlays interior structure of a leg (Argotti et al. 2001) Overlays X-ray image of body (Navab et al. 1999a)	Inset window can be used to show synthetic view of scene with all relevant objects. Context objects should maintain relationship with relevant parts of augmentation, despite error.
Meta-Object	Used to generate objects that do not exist in the real-world, but help solve communicative goals.	Uses virtual arrows (Bryson et al. 1997; Feiner et al. 1993; Klinker et al. 2001; Ogi and Hirose 1996; Reiners et al. 1998) Uses dotted lines to show suggested shots in a pool game (Starner et al. 1998) Virtual markers to show a route taken by a user (Newman et al. 2001) Virtual pipes drawn to show possible locations (Navab et al. 1999b) Virtual cylinder to guide a needle (Sauer et al. 2000; Vogt et al. 2002)	Analogous issues to Include, to show relationship of meta-object to physical world.

Each strategy is defined and examples from AR prototypes are given; if no good example exists, possible uses are suggested. Strategies for adapting to registration error are suggested for each style strategy.

Our approach to dealing with registration error is to use the current error estimates to modify the selection of the style strategies used to achieve a communicative goal. When there is a small-to-moderate amount of registration error, we include style strategies designed to help the user understand the relationship between the augmentation and the physical world.

9.4 Strategies for Visual Context

Many of the graphical objects used to implement the examples in section 9.6 are similar to those used in IBIS, KARMA and other 3D graphical explanation systems (e.g., labelled arrows, wire-frame outlines of objects, 2D inset windows). In this section, we will focus on the strategies that we have used to provide the user with sufficient context to understand the relationship between the augmentations and the physical world.

As mentioned in section 9.1, registration requirements are not absolute: they depend on the domain, the specific context, and the communicative intent of the augmentation. Even for a particular system in a fixed domain, the requirements can change from moment to moment, based on the current augmentation and the physical context in which it occurs. We have designed two visualizations aimed at providing users with sufficient visual context to allow them to understand the intent of the augmentation in the presence of registration error. Both visualizations adapt to the current registration error by showing more detail as the error increases. It is important to note that the amount of registration error that one could tolerate in a system would be application specific as well as augmentation specific.

The KARMA examples in section 9.1 illustrate the two major classes of context we wish to convey to the user. First, our system provides *general visual context* of an augmentation in the physical world, so that a user can understand (roughly) what the intended target of an augmentation is. For example, to provide the general context for the lever on the printer, we would ensure that the printer lid (which contains the lever) is recognizable. Our current approach is to highlight features of the parent object, and show more feature detail as the registration error estimate increases (refer section 9.4.1). In Figure 9.2.b the corners of the computer are highlighted, making the structure of the computer and the location of the power button apparent. Since there is only one button on the right side of the front of the computer, this coarse visualization is sufficient, but the precise location of the button is hard to determine. However, this visualization would be insufficient if there were multiple possible targets of an augmentation, such as the ports on the back of the computer in Figure 9.3.

The second visualization is designed to address this problem by showing the *detailed visual relationships* between an augmentation and other nearby objects in the physical world. By rendering representations of a unique collection of nearby objects, a user can differentiate between the augmentation target and other parts of the physical world that are similar. Returning to the examples in section 9.1, if

there were multiple levers on the printer, and the augmentation target is below a large button, highlighting the levers and the button would reinforce their relationship and allow the user to select the correct lever. Our current approach is to highlight a unique collection of objects near the target of the augmentation in the physical world which highlights are visible on the display, highlighting more objects as the registration error increases (refer section 9.4.2). This visualization allows the user to easily distinguish between the various ports on the back of the computer in Figure 9.3 by displaying nearby details.

The usefulness of the additional visual context relies on the graphical objects being visible to the user. When an insufficient amount of the augmentation is visible to the user, our system uses an inset window to present the augmentation to the user. The inset window contains a complete 3D rendering of the physical objects, with the augmentation overlaid on it. (We currently measure visibility by projecting the bounding box of the objects onto the display. While this approach is sufficient for our current demonstration system, we will be implementing a more general visibility algorithm that handles self-occlusion in the future.)

Unlike previous systems such as IBIS, the design of each augmentation (such as choosing to use an inset window or not) reacts to changing registration error estimates. Therefore, even if the augmentation is visible on the display, when the registration error is large enough, the inset window will be displayed. This is necessary because with a large registration error, there is no guarantee that the augmentation is near the corresponding physical objects, so the physical objects may not be visible even if the augmentation is.

9.4.1 The General Visual Context Visualization Strategy

This visualization draws an increasingly detailed set of edge features of the target object of the augmentation, to give the user a frame of reference to situate the augmentation relative to the physical object, without obscuring the object itself.

Our implementation requires the programmer to define a collection of *feature points* for an object, and the edges that connect to each point. For example, in our examples in Figure 9.4, the eight corners of the computer are defined as feature points, with three edges connected to each corner. When the registration error is small, only a small part of each edge is drawn (e.g., Figure 9.4.b). As the registration error increases, the system draws more of each edge to make the structure of the target object clearer, until the complete edges are drawn (e.g., Figures 9.4.b and 9.4.c).

9.4.2 The Detailed Visual Relationships Visualization Strategy

This visualization highlights a progressively larger set of objects “near” the target of the augmentation, to help the user disambiguate the target of an augmentation from other similar objects.

Our current implementation requires the programmer to define a collection of *feature objects* for an augmentation. Each object is given a priority level corresponding to the order in which they should be added to the scene. The priorities should be based on the uniqueness of the object and when this object should appear. For example, for small registration errors, the augmentation and feature objects will be very close to their actual physical locations, so the programmer may wish to give higher priority to objects that are farther away from the target of the augmentation to avoid obscuring it. As error increases, objects closer to the augmentation target can be used with less likelihood of them obscuring the target of the augmentation. The feature objects are stored in a priority queue, and as registration error increases, features are removed from the queue and added to the scene according to their priority.

Our implementation has intentional hysteresis. In order to maintain temporal coherence, it is important that augmentations do not flicker in and out of a scene. Therefore, once a feature is drawn, it will remain in the scene until registration improves significantly.

9.4.3 Using Inset Windows to Reduce Ambiguity

As discussed above, an inset window is used to show the augmentation when the target is not visible, or when the registration error is sufficiently large that the augmentation and the target may not be visible at the same time. We also use the inset window in other situations, as illustrated in Figure 9.4. In this example, the scene contains a virtual object, the ghosted representation of the CD tray, which is being used to show the change that occurs when the CD button is pressed. In this case, when there is more than a very small amount of registration error, we use the inset window to show this change and remove the virtual CD tray from the augmentation. As context objects are added to the scene, we wish to limit the number of objects in the augmentation to reduce the possible ambiguity if they happen to appear similar to objects in the physical world.

As with the context objects in section 9.4.2, we do not allow the inset window to flicker rapidly in and out of the scene as the augmentation moves around. Instead, it slowly fades into and out of the scene as necessary.

9.5 Implementation

The version of the AIBAS system described in this chapter was implemented in Java and Java3D on a Sun Ultra 60 workstation. The core part of the system (that computes which features are visible, activates and displays the inset window, etc.) was implemented as a library that was then used to create the four example augmentations shown in Figures 9.1 to 9.4. We are currently re-implementing the system in C++, using the OpenSceneGraph graphics library.

The illustrations in this chapter were not rendered on an HMD using a live tracking system. Rather, to facilitate experimentation, we used a static image background and received position reports from a simulated tracker that lets us simulate error using a 2D graphical interface. The simulated tracker takes a single fixed location, adjusts it based on the error, and reports it to our test programs. The thresholds at which the augmentation changes are programmer defined. In Figures 9.1 to 9.4, we have zoomed in on specific areas of the scene in order to show clear views of our context strategies being applied; the user is not moving.

The cutaway view in Figure 9.1 is implemented by cutting a rectangular hole in a simple model of the computer, in real time, to create an exterior model with a cutout. By controlling the rendering order, we can render the exterior model into the z-buffer, followed by the interior model, so that the interior shows through the hole without obscuring the video image of the real computer. In the future, we intend to create a more stylized cutout border using a general CSG package.

9.6 Examples

In this section, we give four examples of how our context visualizations can be used. Each example implements a specific communicative goal for a hypothetical set of maintenance tasks on a computer, and represents a broad class of augmentations common in AR systems. In each example, we first show the augmentation with no registration error (part (a)), then show it with small-to-moderate error (part (b)), and finally show the augmentation with a larger error (part (c)). In each case, the transition between these augmentations is handled automatically by the system, and is based on both the amount of registration error and the visibility of the target of the augmentation. In each example, we display the augmentation when the error is very large (which typically appears far from the actual computer in the scene) to show the reader the degree of registration error; however, the augmentation would be disabled when the error is this large.

9.6.1 Augment a Hidden Object

In this example, the goal is to show the graphics card located inside a workstation. The idea of using AR to give a user “X-ray” vision is one of the goals of many AR system prototypes.

When registration is perfect, we make the graphics card visible by using a cutaway view through the side of the case (Figure 9.1.a). When registration error is small-to-moderate, the computer case is used as the basis for the *general visual context* visualization, causing the corners of the computer to be drawn (Figure 9.1.b). Since the user cannot see the card, drawing additional context near the card provides no benefit. When registration error is large, the card is shown using an inset window that renders a cutaway view of the graphics card at the correct position on a 3D model of the computer.

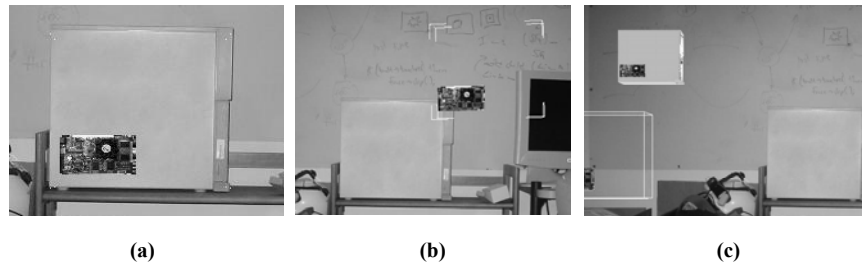


Fig. 9.1. Augment a hidden object, **(a)** Perfect registration. Shows the computer graphics card located inside of the computer. A cutaway view is used to make the graphics card visible, **(b)** Moderate registration error. The edges of the computer are used to show the relative location of the graphics card within the computer, **(c)** Significant registration error. An inset window is drawn to show the intended location of the augmentation. (The augmentation is shown for illustrative purposes only; it would normally be removed at this point)

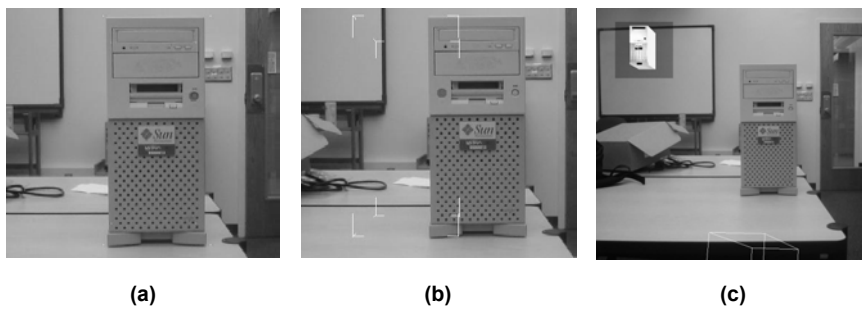


Fig. 9.2. Augment a unique visible object, **(a)** Perfect registration. The power button on the front of the computer is highlighted. The colour red is used to draw focus to the power button, **(b)** Moderate registration error. The edges of the computer are used to show the relative location of the power button on the front of the computer, **(c)** Significant registration error. An inset window is drawn to show the intended location of the augmentation. (The augmentation is shown for illustrative purposes only; it would normally be removed at this point)

9.6.2 Augment a Unique Visible Object

In this example, the goal is to show the location of the power button on the front face of a workstation. Pointing at, highlighting and otherwise augmenting visible objects is a common goal in AR system prototypes. As is often the case, the object is unique. There are no other buttons on the computer, so there is no confusion as to which button we are trying to highlight if registration is not perfect.

When registration is perfect, we highlight the power button by tinting it red to draw the user's focus to it (Figure 9.2.a). When registration error is small-to-

moderate, the computer case is used as the basis for the *general visual context* visualization, causing the corners of the computer to be drawn (Figure 9.2.b). Because of the simplicity of the front of the workstation, there is no need to draw any additional features in the scene. When the error is large or the augmentation is off screen, an inset window is used to make the highlight of the button visible over a 3D model of the computer (Figure 9.2.c).

9.6.3 Augment an Ambiguous Visible Object

In this example, the goal is to visualize the location of one of the serial ports on the back of the workstation. There are many ports and plug-ins in the back that could be confused for the intended serial port if the augmentation is misregistered. Augmenting ambiguous objects, such as parts of a machine, is also common in AR systems.

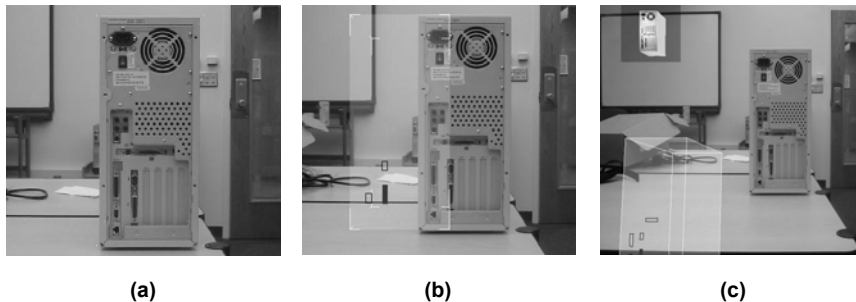


Fig. 9.3. Augment an ambiguous visible object, **(a)** Perfect registration. Shows a particular port on the back of the computer being highlighted. The colour red is used to draw focus to the port, **(b)** Moderate registration error. The edges of the computer and ghosting are used to show where the ports are located on the back of the computer, and nearby features (other ports) are drawn to clarify which port is of interest, **(c)** Significant registration error. An inset window is drawn to show the intended location of the augmentation. (The augmentation is shown for illustrative purposes only; it would normally be removed at this point)

When registration is perfect, we highlight the port by tinting it red to direct a user's focus to it (Figure 9.3.a). When registration error is small-to-moderate, we add general context, as above, and apply the *detailed visual relationships* visualization to the visible objects around the port to show the relationship between the intended port and those other objects on the back of the computer. Each of these object features is highlighted by rendering its silhouette in black. As error gets worse, more features are added (Figures 9.3.b and 9.3.c). As above, when the augmentation is no longer in the field of view, or the error is large, an inset window is used to show the location of the port on a 3D model of the computer.

In this example, we also add a ghost of the computer as part of the general visual context visualization. This ghost becomes more opaque as the error increases, although it remains very transparent. (While we are not yet sure if this added information is necessary, we show it here because it was a common suggestion given to us when we demonstrated the system to visitors to our laboratory.) We hope to determine experimentally if the ghost helps, hinders or does not affect a user's ability to understand the augmentation (refer to section 9.8).

9.6.4 Augment a Scene with Virtual Objects

In this example, the goals are to visualize the action of pushing the CD tray button to open the tray, and the movement of the tray that results from this action. The button is beside a similar button on the CD drive. Additional objects are added to the scene to accomplish these goals: a labelled arrow, a label and leader line to the button, and a 3D model of the CD tray opening. Many AR systems include obviously synthetic objects (like labels and arrows) and models of real objects that have a specific relationship to the world.

A synthetic red arrow labelled "push" is included to show the action to take (push the button), and the button is highlighted with a red silhouette. Red is used to draw the user's attention to the CD tray button. Another synthetic object, in this case a label with a leader line identifying the button, is included for clarity. With perfect registration, a ghosted CD tray is included in the scene to show the result of pushing the button (e.g., the tray opening and closing in Figure 9.4.a). When registration error is small-to-moderate, we add general context as above, and add object features (the neighbouring button and the CD tray door) to show the detailed visual context around the button (Figures 9.4.b and 9.4.c), and show two different states of the visualization with slightly more error in Figure 9.4.c.

Unlike previous examples, when the error is small-to-moderate we replace the virtual CD tray with an inset window containing a 3D model of the computer showing the tray opening and closing. The reason for this is twofold. First, since the virtual CD tray does not correspond to any real object, it may be confused with the general and detailed visual context objects that have been added to the scene. Second, in this case, the CD tray model blocks one of the two feature objects being used by the context strategy, rendering it useless. Finally, when registration error is large, the synthetic objects and the highlight are moved to the inset window (not shown).

9.7 Discussion

In this chapter, we have revisited the conceptual framework of *communicative intent*, and use it as the basis for providing semantic knowledge of the graphics generated by AR systems. Such a system is especially applicable in industrial

settings, where detailed information about equipment and processes is available. We have discussed both how intent can be useful in AR, and especially how knowledge of the intent can allow an augmentation to be modified to ameliorate the effects of registration error. We have implemented two visualization techniques to convey both general and specific context to the viewer. Our simple implementation, and four demonstration prototypes for a maintenance task, shows how this approach is a promising first step toward creating AR systems that function with “good enough,” rather than perfect, tracking and registration. Such systems are ideal for enhancing manufacturing tasks such as assembly, maintenance and repair.

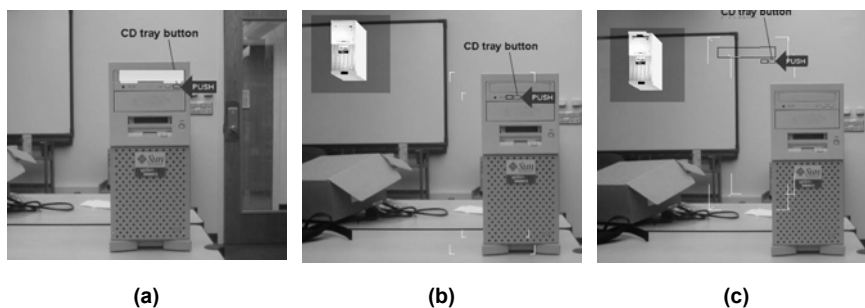


Fig. 9.4. Augment a scene with virtual objects, **(a)** Perfect registration. The CD tray button on the front of the computer is highlighted, an animated CD tray opens and closes to show what happens when the button is pushed, an arrow points to the CD tray button telling the user to “push,” and a label identifies the CD tray button. The colour red is used for the outline and the arrow to draw focus to the CD tray button, **(b)** Small registration error. The edges of the computer are used to show the relative location of the CD tray button on the front of the computer and a feature (an outline of another button) is drawn in order to give context. The CD tray animation is moved to an inset window so it does not interfere with the other augmentations or the user’s perception of the context objects. (The CD tray in the inset is closed.), **(c)** Moderate registration error. A second feature (the CD tray door) is added to the context. (The CD tray in the inset is open.)

9.8 Future Work

We are continuing to develop these tools and the ideas behind them, and integrate them into the new AR systems we are developing in our laboratory. In particular, the two techniques for displaying context could be made more robust, including future automation of the selection and rendering of feature points and objects.

There are a number of other areas that we feel are ripe for continued work, some of which we expect to begin working on soon:

User experiments: As previously mentioned, we are designing a set of experiments to test the utility of this approach, and of our current visualizations. Our first experiment, which we are currently designing, will have subjects perform

a construction task using building blocks of various shapes and sizes. We will vary the amount of registration error (added to an otherwise accurate tracker), and compare performance with and without our augmentations.

Manipulating augmentations: Another technique we would like to investigate involves monitoring the registration error of augmentation target objects, and moving as much of the augmentation onto the screen as possible, given the possible locations of the physical object. While we could deal with this situation by displaying the augmentation in an inset window, in certain situations it might be useful to move the augmentation to a plausible onscreen location so the user sees it. This would provide a better view of the augmentation, but has the drawback of potentially doubling the registration error.

Manipulating video: Another possible addition to our list of techniques involves manipulating the real-time video that is placed into the system as a background. For example, the system could zoom into a particular area of the video and use it as the background of an inset window. While the image would be grainy from being scaled up, any augmentations overlaid on it could take advantage of the increased resolution. Image processing techniques could also be used to manipulate the video around the location of a physical object in order to highlight it (e.g., by brightening the pixels) or de-emphasize it (e.g., by dimming the pixels).

Adapting to ambient colour: Many augmentations are intended to blend into the world (e.g., virtual objects added to a scene), while some work best if they stand out from their surroundings (e.g., maintenance instructions). For the latter, we are considering examining the video image to find colours to use for augmentations that are as different as possible from the video of the world. This technique could also work for optically see-through displays as well, if the user is wearing a camera.

9.9 Conclusions

This research demonstrates that AR systems do not need perfect tracking to generate understandable augmentations. By creating visualizations that adapt to registration error, we hope to support the development of AR systems that can be useful and effective in the presence of imperfect tracking. We have demonstrated a visualization system that points the way toward accomplishing this goal.

Acknowledgements

This work was supported by Siemens via a GVU Industrial Affiliate Grant, and by the Office of Naval Research under Grant N000140010361. We would also like to thank Enylton Coelho for his help with the implementation of the prototype system.

References

- Andre AD, Cutler HA (1998) Displaying uncertainty in advanced navigation systems. In: Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting, Chicago, IL, pp 31-35
- Argotti Y, Davis L, Outters V, Rolland J (2001) Dynamic Superimposition of Synthetic Objects on Rigid and Simple-deformable Real Objects. In: Proceedings of International Symposium on Augmented Reality, New York, New York, pp 5-10
- Azuma R, Bishop G (1994) Improving static and dynamic registration in an optical see-through HMD. In: Computer Graphics (Proc. ACM SIGGRAPH '94), Annual Conference Series, Orlando, FL, pp 197-204
- Bajura M, Fuchs M, Ohbuchi R (1992) Merging Virtual Objects with the Real World: Seeing Ultrasound Imagery within the Patient. In: Proceedings of SIGGRAPH '92 (Chicago, IL, July 26-31, 1992). Computer Graphics 26(2):203-210
- Bell B, Feiner S, Höllerer T (2001) View Management for Virtual and Augmented Reality. In: Proceedings of ACM Symposium on User Interface Software and Technology, Orlando, FL, pp 101-110
- Billinghurst M, Bowskill J, Dyer N, Morphet J (1998) An Evolution of Wearable Information Spaces. In: Proceedings of Virtual Reality Annual International Symposium. Atlanta, Georgia, pp 20-27
- Bryson S, Johan S, Schlecht L (1997) An Extensible Interactive Visualization Framework for the Virtual Windtunnel. In: Proceedings of Virtual Reality Annual International Symposium, Albuquerque, New Mexico, pp 106-113
- Darken R, Cevik H (1999) Map Usage in Virtual Environments: Orientation Issues. In: Proceedings of IEEE Virtual Reality, Houston, Texas, pp 133-140
- Dinsmore M, Langrana N, Burdea G, Ladeji J (1997) Virtual Reality Training Simulation for Palpation of Subsurface Tumours. In: Proceedings of Virtual Reality Annual International Symposium, Albuquerque, New Mexico, pp 54-60
- Feiner S, MacIntyre B, Seligmann D (1993) Knowledge-based Augmented Reality. Communications of the ACM 36(7):52-62
- Feiner S, Webster A, Krueger T, MacIntyre B, Keller E (1995) Architectural anatomy. Presence 4(3):318-325
- Feiner S, MacIntyre B, Höllerer T, Webster A (1997) A Touring Machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. Personal Technologies, 1(4):208-217
- Holloway RL (1997) Registration Error Analysis for Augmented Reality. Presence: Teleoperators and Virtual Environments 6(4):413-432
- Furmanski C, Azuma R, Daily M (2002) Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information. In: Proceedings of International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, pp 215-225
- Gooch A, Willemsen P (2002) Evaluating Space Perception in NPR Immersive Environments. In: Non-Photorealistic Animation and Rendering. Annecy, France, pp 105-110
- Grimson WEL, Ettinger GJ, White SJ, Lozano-Perez T, Wells WM III, Kikinis R (1996) An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. IEEE Transactions on Medical Imaging 15(20):129-140

- Hoff W (1998) Fusion of data from head-mounted and fixed sensors. In: Proceedings of the First International Workshop on Augmented Reality, San Francisco, CA, pp 167–182
- Interrante V, Fuchs H, Pizer S (1995) Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. In: Proceedings of IEEE Visualization, Atlanta, GA, pp 52-59
- Johnson A, Moher T, Ohlsson S, Gillingham M (1999) The Round Earth Project: Deep Learning in a Collaborative Virtual World. In: Proceedings of IEEE Virtual Reality, Houston, Texas, pp 164-171
- Julier S, Lanzagorta M, Baillet Y, Rosenblum L, Feiner S, Höllerer T, Sestito S (2000) Information Filtering for Mobile Augmented Reality. In: Proceedings of International Symposium on Augmented Reality, Munich, Germany, pp 3-11
- Kanbara M, Okuma T, Takemura H., Yokoya N (2000) A Stereoscopic Video See-Through Augmented Reality System Based on Real-time Vision-based Registration. In: Proceedings of IEEE Virtual Reality, New Brunswick, New Jersey, pp 255-262
- Klinker G, Creighton O, Dutoit AH, Kobylinski R, Vilsmeier C, Brugge B (2001) Augmented maintenance of powerplants: a prototyping case study of a mobile AR system. In: Proceedings of International Symposium on Augmented Reality, New York, New York, pp 124-133
- Kutulakos K, Vallino J (1996) Affine Object Representations for Calibration-Free Augmented Reality. In: Proceedings of Virtual Reality Annual International Symposium, Santa Clara, California, pp 25-36
- MacIntyre B, Coelho E, Julier S (2002) Estimating and Adapting to Registration Errors in Augmented Reality Systems. In: Proceedings of IEEE Virtual Reality, Orlando, Florida, pp 73-80
- Maren AJ, Ali M (1988) Hierarchical Scene Structure Representations to Facilitate Image Understanding. In: Proceedings of the First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, vol 2, pp 657-667
- Moezzi S, Katkere A, Kuramura D, Jain R (1996) Immersive Video. In: Proceedings of Virtual Reality Annual International Symposium, Santa Clara, California, pp 17-24
- Navab N, Bani-Hashemi A, Mitschke M (1999a) Merging Visible and Invisible: Two Camera-Augmented Mobile C-arm (CAMC) Applications. In: Proceedings of International Workshop Augmented Reality, San Francisco, California, pp 134-141
- Navab N, Bascle B, Appel M, Cubillo E (1999b) Scene Augmentation via the Fusion of Industrial Drawings and Uncalibrated Images with a View to Maker-less Calibration. In: Proceedings of International Workshop Augmented Reality, San Francisco, California, pp 125-133
- Newman J, Ingram D, Hopper A (2001) Augmented Reality in a Wide Area Sentient Environment. In: Proceedings of International Symposium on Augmented Reality, New York, New York, pp 77-86
- Ogi T, Hirose M (1996) Multisensory Data Sensualization Based on Human Perception. In: Proceedings of Virtual Reality Annual International Symposium, Santa Clara, California, pp 66-71
- Piaget J, Inhelder B (1956) *The Child's Conception of Space*. Humanities Press, New York
- Reiners D, Stricker D, Klinker G, Muller S (1998) Augmented Reality for Construction Tasks: Doorlock Assembly. In: Proceedings of International Workshop Augmented Reality, San Francisco, California, pp 31-46

- Risch J, May R, Thomas J, Dowson S (1996) Interactive Information Visualization for Exploratory Intelligence Data Analysis. In: Proceedings of Virtual Reality Annual International Symposium. Santa Clara, California, pp 230-238
- Rock I (1975) *An Introduction to Perception*, MacMillan Publishing Co., New York
- Sauer F, Wenzel F, Vogt S, Tao Y, Genc Y, Bani-Hashemi A (2000) Augmented Workspace: designing an AR testbed. In: Proceedings of International Symposium on Augmented Reality, Munich, Germany, pp 47-53
- Seligmann D (1993) *Interactive Intent-Based Illustration: A Visual Language for 3D Worlds*. Ph.D. thesis, Columbia University
- Seligmann D, Feiner S (1991) Automated Generation of Intent-Based 3D Illustrations. *Computer Graphics*, 25(4), July 1991 In: Proceedings of ACM SIGGRAPH, Las Vegas, Nevada, pp 123-132
- Starner T, Schiele B, Rhodes B, Jebara T, Oliver N, Weaver J, Pentland A (1998) Augmented Realities Integrating User and Physical Models. In: Proceedings of International Workshop Augmented Reality, San Francisco, California, pp 73-80
- Simon G, Berger M (2002) Reconstructing while registering: a novel approach for markerless augmented reality. In: Proceedings of International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, pp 285-294
- State A, Livingston MA, Garrett WF, Hirota G, Whitton MC, Pisano ED, Fuchs H (1996) Technologies for Augmented Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. In: Proceedings of ACM SIGGRAPH, New Orleans, Louisiana, pp 439-446
- Sung U, Yang J, Wohn K (1999) Concurrency Control in CIAO. In: Proceedings of IEEE Virtual Reality 1999, Houston, Texas, pp 22-28
- Tang A, Owen C, Biocca F, Mou W (2003) New techniques for presenting instructions and transcripts: Comparative effectiveness of augmented reality in object assembly. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, Fort Lauderdale, Florida: ACM Press, pp 73-80
- Vogt S, Khamene A, Sauer F, Niemann H (2002) Single Camera Tracking of Marker Clusters: Multiparameter Cluster Optimization and Experimental Verification. In: Proceedings of International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, pp 127-136
- Woods D (1995) Toward a theoretical base for representation design in the computer medium: Ecological perception and aiding human cognition. In: Flack, Hancock, Caird, Vicente K (eds) *An Ecological Approach to Human-Machine Systems I: A Global Perspective*. Lawrence Erlbaum Associates, pp 157-188
- Zusne L (1970) *Visual Perception of Form*. Academic Press, New York London