Proceedings of the ASME 2011 World Conference on Innovative Virtual Reality
WINVR2011
June 27-29, 2011, Milan, Italy

WINVR2011-5569

TABLE TOP AUGMENTED REALITY SYSTEM FOR CONCEPTUAL DESIGN AND PROTOTYPING

Michael VanWaardhuizen
Virtual Reality Applications Center
Iowa State University
Ames, IA, USA

James Oliver
Virtual Reality Applications Center
Iowa State University
Ames, IA, USA

Jesus Gimeno
Institut de Robotica I de Tecnologias de la Informacion y des Communicions
Valencia, SPAIN

ABSTRACT

The AugmenTable is a desktop augmented reality workstation intended for conceptual design and prototyping. It combines a thin form factor display, inexpensive web cameras, and a PC into a unique system that enables natural interaction with virtual and physical parts. This initial implementation of the AugmenTable takes advantage of the popular open source augmented reality software platform ARToolkit to enable manual interaction with physical parts, as well as interaction with virtual parts via a physically marked pointer or a color-marked fingertip. This paper describes similar previous work, the methods used to create the AugmenTable, the novel interaction it affords users, and a number of avenues for advancing the system in the future.

INTRODUCTION

Among the most dramatic trends emerging over the past several years include the use of more natural, direct interfaces, the rise of consumer-grade mixed reality systems, and the application of virtual reality to design and manufacturing processes. These trends imply that users will soon need 3D interfaces to interact with technology. This paper describes a unique project that attempts to unify these trends and results in an augmented reality workstation that allows users to interact with 3D virtual objects directly with their bare hands.

DIRECT MANIPULATION INTERFACES

Direct manipulation interfaces, also known as natural interfaces, are those that require few or no mediating controls for interaction [1]. For example, multitouch displays, like those found in Apple iPads, allow the user to touch application content and controls directly with his or her fingertip rather than using a mediating technology like a keyboard or mouse.

Direct manipulation interaction has many benefits, most notably a decreased need for training or practice in order for a user to expertly operate the interface – in fact, humans have evolved to intuitively manipulate objects with their hands. These benefits often translate into easier, more attractive, and more successful designs. For example, Reifinger et al. [2] show that direct hand manipulation of virtual objects is faster and more intuitive than using a keyboard/mouse interface. These benefits have led entire research groups, such as MIT's Tangible Media Group, to dedicate more than 10 years to integrating manipulatable objects with virtual objects and metadata.

A number of consumer technologies, both nascent and established, aim to increase the prevalence of direct manipulation interfaces. Multitouch displays allow finger presses directly on a screen and have become ubiquitous in smartphones and other displays. Other technologies that have not seen the same market penetration include devices such as desktop haptic manipulators (e.g., Phantom Omni), 3D pointers or instrumented gloves. These devices may become more common and less expensive as user's expectations for direct manipulation interfaces rise.

AUGMENTED/MIXED REALITY

Another trend is the expansion of augmented reality systems. Broadly, augmented reality is the superposition (most frequently visual) of real and virtual objects or information in one environment. As a research area, augmented reality has been pursued for many years with a number of wide-ranging applications. Many of these systems have never left the laboratory due to cost or other constraints rendering them impractical. However, due to the adoption of mobile devices with powerful processors, built-in cameras, and fast internet connections, augmented reality is beginning to infiltrate the average individual's life.

A number of augmented reality applications have appeared in the Apple and Google application stores (see [3] or [4] for examples.) These applications range from spur-of-the-moment information overlays, like location guides, reviews and ratings,

to games that observe the user's motions to create virtual effects. One good example is Google's Goggle program [5], an application that accepts photos of landmarks, books, artwork, and many other object types and then returns a Google visual search on the object.

As the public uses of augmented reality are accelerating, so are the technologies that power them. Many examples of improved augmented reality applications are here or on their way. MIT's Sixth Sense demo combines an iPhone, video camera, and pico-projector to allow a user to record and display on any surface [6]. The Skinput system creates a similar effect using the user's skin as an input device [7]. Other consumer technologies such as Samsung's transparent OLED displays [8] will one day enable a generation of hands-off, information-everywhere augmented reality. This trend has only just begun.

VR IN DESIGN/MANUFACTURING

The trend of using virtual and augmented reality to support design and manufacturing processes is not one that receives significant attention from the general public, yet is a source of new thinking about what problems VR/AR can solve. Though many systems are proprietary, a number of design/manufacturing AR systems have been described in academic papers. Kim and Dey, for example, discuss the use of augmented reality for design prototyping activities [9]. Augmented and virtual reality provides the next extension to current computer-aided design systems, providing a means to more in-depth conceptual design, review, and prototyping.

Academic literature also provides several guidelines for industrial augmented reality systems. Kim and Dey claim that immersive displays such as head mounted devices (HMDs) are important to reach the full capability of an industrial AR system. Additionally, Bleser, et al. [10], state that the use of markers for hand tracking systems is not acceptable for industrial applications. These criteria create a necessity for a new industrial AR design.

USER NEEDS

These trends have two co-dependent sources: technological innovation to create business opportunities, and the creativity of developers to meet real user needs with technology. However, the ongoing growth of these trends is driven more by consumer and user adoption. Users seek direct manipulation because it is quicker, easier, and more pleasant to use. Consumers are using more augmented reality because it is becoming inexpensive and requires less expertise or preparation. VR is becoming more important to design and manufacturing because it is providing new means of creating designs and analyzing them.

To capitalize on these trends a new system should have these same properties. It should be simple to use – it should not require learning a gestural language or require the user to wear or manipulate cumbersome equipment. It must provide a new way of interacting with virtual (or real) objects to enable new perspectives. Finally, it must be an inexpensive system that can be assembled without great expertise. These are the requirements for a system to effectively provide value to users.

THE AUGMENTABLE

The system described in this paper provides an immersive augmented reality environment that enables a direct manipulation interface for a conceptual design process and enables new human-computer interaction. The system, called the AugmenTable and shown in Figures 1 and 2, is a desktop-based workstation that features inexpensive cameras, a thin display monitor (to approximate a transparent view of the desktop), established computer vision algorithms to identify and track a user's hands, and virtual affordances for a user to manipulate or interact with a virtual object using his or her bare hands. Furthermore, the system interaction is intended to provide direct manipulation with virtual objects that is inherently similar to the way that user's interact with real objects. This similarity enables a greater sense of immersion and suggests a number of interaction metaphors that can be directly copied from everyday life. As a result, this system is intended to provide a test-bed for future research into three-dimensional hand-based interactions.

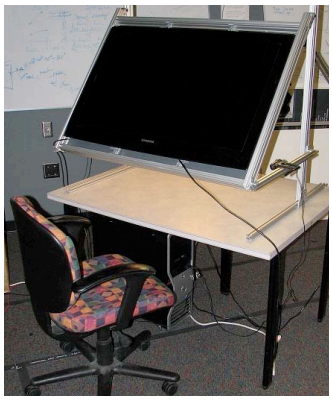

Figure 1: AugmenTable - Front

Figure 2: AugmenTable - Back

The apparatus provides this functionality without encumbering the user with wearable equipment. The system uses cameras and computer vision to track the hands without requiring gloves or ungainly makers. The display provides a view of the user's hands and virtual objects integrated together without necessitating a bulky HMD.

To be effective, the system was designed under a set of constraints: the system had to be real-time and avoid noticeable lag (defined by von Hardenberg and Bérard as a maximum update interval of 50ms, equating to a refresh rate of 20 Hz [11]), and had to be flexible to support a variety of application designs including multi-person collaboration. An additional requirement was for the system to be relatively inexpensive to encourage adoption.

This paper describes previous work towards these goals, the methods used to realize it, the strengths and weaknesses of the AugmenTable, and future work.

RELATED WORK

The combination of augmented reality and gesture interaction is not a new goal. Many systems over the past 15 or more years have aimed to provide more natural interaction in

virtual environments via gesture recognition. Each system that has been developed has its own strengths and limitations. Here, significant previous work is reviewed in hand tracking, gesture interaction, and augmented reality. Emphasis is given to research published within the last ten years.

AUGMENTED REALITY

Augmented reality is the blending of sensory input from the "real world," most typically visual information acquired from cameras or the user's own eyes, and virtual sensory input. The virtual input can range from textual or visual information to 3D geometry such as guiding arrows or virtual objects. Most augmented reality systems today are based on computer vision techniques that identify preset markers (preregistered 2D images) in a camera image, calculate the marker's position and attitude, and then superimpose the virtual inputs in the viewing stream.

This paper foregoes a thorough review of augmented/mixed reality literature in favor of examining integrated systems. A definitive bibliography can be found in the ACM SIGGRAPH Asia 2008 course documentation.

GESTURE INTERACTION

Gesture interaction with computers also has a long history. Gesture interaction poses two problems: how to observe or track the hands, and how to translate the hand's position, attitude, or motion into computer interaction.

Hand Tracking

Tracking of the hands is accomplished in one of two ways: applying some form of external accessory to the hands that is easily tracked (also called instrumented hands), or using computer vision algorithms and techniques to extract hand information from one or more cameras.

One form of accessory is fiducial markers. Fiducial markers are preset images typical of augmented reality systems such as those using HIT Lab's ARToolkit [12]. An example system using such markers is FingARTips [13]. This system requires users to wear a black glove adorned with several markers at important joints (see Figure 3). The markers are then detected in an AR environment, allowing several direct manipulation interactions such as pressing, pointing, and grabbing. The use of fiducial markers for tracking reduces the complexity of the tracking system. However, it also limits the range of motion of the hands because the markers must be visible at all times, restricting the angles and rotation of the hands in 3D. A similar marker-based gestures system was used by Kato et al. for collaborative interaction as well [14].

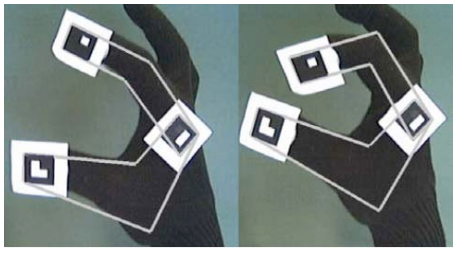

Figure 3: Hand Model of FingARTips [13]

Markers are not limited to fiducial markers for AR. Reifinger et al., created a system using small markers on a glove that were tracked by infrared cameras, with a scene displayed via HMD [2]. This system is able to recognize both static and dynamic gestures via a hidden Markov model. The system supported grasping and scaling manipulations similar to the AugmenTable, but required unwieldy IR markers and specialized cameras to do so. These requirements imposed a high cost and reduced the immersiveness of the application.

Markers can provide useful information about the articulation of the hand, and so are often used in systems that create a computational model of hand geometry. Such information can also be gained from sensor equipped "datagloves." An older review of glove based inputs was performed by Sturman and Zeltzer [15]. More simply, colored gloves (as used by Keskin et al., [16]) are much less encumbering and have been used to address difficulties with skin detection.

Another unique finger tracking solution was established by Walairacht et al., [17]. They describe a system where a user may manipulate a virtual object in a workspace with very real, natural hand movements. The system provides haptic feedback, enabling the user to touch and manipulate objects as if they were real. The system also tracks all of the user's fingers individually, allowing for geometric calculation of the user's perspective. This is accomplished through the use of a unique system of strings, attached to the user's hands during operation, as shown in Figure 4. This system, though enabling many capabilities, would not be practical for casual use and could likely result in a fair amount of user fatigue. Additionally, the system required numerous calculations, which resulted in a lagged, slow-system response.

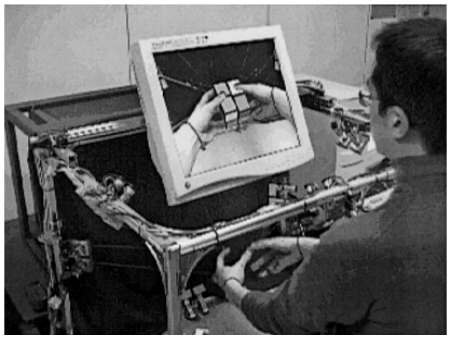

Figure 4: SPIDAR-8 Haptic AR [17]

When all encumbrances are removed, hand tracking is the province of computer vision techniques. The number of papers and techniques developed are many and myriad. Most systems utilize skin detection and object tracking algorithms. Unfortunately, there is no "silver bullet" technique commonly accepted to detect hands. Each technique addresses some difficulties at the expense of others.

Markerless tracking of hands is not a new idea. DigitEyes was one of the earliest markerless hand tracking systems described in 1994 [18]. Four years later, Nölker and Ritter advocated markerless realtime hand tracking without using a geometric model to improve speed [19]. Though markerless tracking has been suggested and used for more than 15 years, marked tracking is still considered justifiable due to the difficulties of markerless hand recognition.

The following systems all place restrictions on either the environment or the user's gestures in order to ameliorate the difficulties of hand tracking. The most common restrictions are uniform background and limited gesture speeds [11]. Other restrictions may be on the orientation of the hand to remove self-occlusions or to limit the tracking of the hand to two dimensions, such as on a desk surface [18], [20], [21], [22], and [23]. These restrictions are often necessary for the systems to function, or may be implicit in the tasks the system supports. However, the more restrictions imposed on the user can render the experience less immersive and less realistic, reducing the value to the end user. As a result, most of these systems never leave the laboratory.

Hand Interaction

Erol et al. [24] categorize hand interactions into two types: gestures used for communication (in this context, to command and control interfaces) and object manipulation gestures (simulating life-like interactions, such as pointing or pinching). The former tends to utilize static hand poses or motion patterns which are then interpreted as commands. The latter may include poses and motion patterns, but also frequently include direct tracking of the hand or fingertips.

Pose and motion pattern recognition is developed either in creating three dimensional models of the hand through inverse kinematics, or in partial pose recognition based on 2D appearance [24]. Model based/inverse kinematic reconstruction is not discussed in this thesis.

Pose recognition is separated into tasks of identifying the hand in one or more images, extracting relevant features, and passing them to a gesture classification system. Such a system uses statistical methods to determine the pose or gesture from a previously trained library. The variants of this technique are very popular for hand interaction, and were used by [11], [21], [25], [26], [27], [28], [29], and [30].

Other interactions arise from the development of native 3D interactions. Natural, realistic hand interactions such as grabbing, pinching, and bumping are new to 3D environments. When these are not possible, another class of interactions use virtual controls or widgets that are designed for 3D interaction. Hand [31] found that well designed widgets can be less damaging to the feeling of directness than more abstract or invasive interfaces like gestures or physical controls.

COMPARABLE SYSTEMS

Several systems have previously been developed with the goal of virtual object manipulation in an augmented reality environment using markerless hand tracking. However, all do not entirely reach the goal of intuitive, unencumbered fingertip manipulation of virtual objects.

The apparatus for such a system is fairly well agreed upon. Erol et al., point out that multiple cameras are necessary for object manipulation without using markers, or for allowing two-handed interactions [24]. They also mention that combining multiple views to establish correspondences across cameras and 3D features has not been explored well. Abe et al., use vertical and horizontally oriented cameras to develop a 3D position of a single finger, enabling 3D rotation and translation when combined with pose recognition based commands [21]. A similar multi-camera system was developed by [32] several years prior. These systems are not augmented reality, though, since they do not integrate real objects with virtual objects.

An early tabletop AR system was developed by Oka et al. called EnhanceDesk [33]. By using a color camera and an infrared camera, fingertips are tracked through a combined approach of template matching and Kalman filtering. This system only tracked the fingers on the surface of a desktop and today would be more effectively implemented through multitouch surfaces. That said, this system's apparatus and methods have been applied to the 3D problem by subsequent systems, including the AugmenTable. A similar system with similar restrictions was more recently suggested in [34].

A more capable system described by Lee and Höllerer shares many of the same goals as this proposed system [35]. Based on previous "HandyAR" work [36] and markerless AR research [37], Lee and Höllerer use an optical flow algorithm to track an outstretched open hand. It determines the finger locations based on the thumb location, then uses pose estimation to determine the orientation of the hand (see Figure 5.) The finger positions are established through an initial calibration, then tracked using Kalman filtering. This enables a coordinate system or model to be matched to the user's hand as though the hand were a 2D fiducial AR marker.

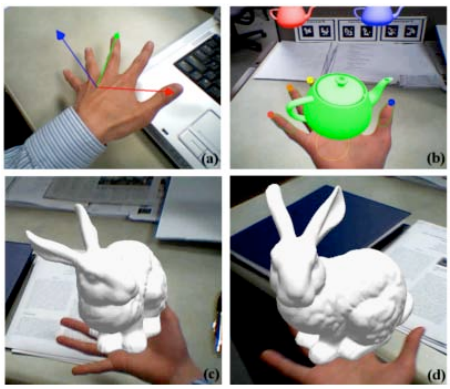

Figure 5: HandyAR Snapshots [36]

The recent extension to this work enabled the tracking of desktop surfaces for an additional AR surface, as well as a "grabbing" gesture through breaking the tracking of the outstretched hand in favor of a closed fist. This allows free manipulation of a virtual object with a user's hands, but at the cost of losing natural gestures such as pointing, grabbing, or pinching that deform the hand. This system is unable to track motion through self-occlusion as well. Rotating a virtual object with the hand can only be done within a limited range of motion. A model cannot be rotated to its side, for example.

Song et al. [38] describe a more effective system. This system tracks an individual finger in a 3D augmented reality environment. A set of interaction methods is created, combined with a physics engine, to provide a unique object manipulation system. The authors also conducted a user study finding bare hand interactions to be more intuitive and pleasant for users than keyboard and mouse interfaces. Using a single finger,

however, is limiting and does not match natural human object manipulation.

Of all the systems and prototypes reviewed, the one developed and described by Kolarić et al., bears the most common ground with the proposed system [39]. Kolarić's system uses free, unmarked hand movements to manipulate virtual 3D objects. They use a computer vision system that tracks the hands in a stereo camera setup and implement the Viola-Jones tracking method paired with skin color histograms for detection. To manipulate objects, the authors define a set of hand poses for command communication: select, open, and closed, which are mapped to functions such as select, translate, and rotate.

This system (shown in Figure 6) bears the same functional purpose as the proposed AugmenTable system. However, the AugmenTable tracks fingertip points for higher controllability, does not use learned hand gestures in favor of developing intuitive manipulation widgets, and uses an apparatus that allows for the hands and virtual objects to inhabit the same perceptual space. Additionally, the proposed system supports multiple hands, multiple fingertips as well as rotation of the hand through arbitrary angles – a rare combination in the field.

All of these comparable systems use either head mounted devices or regular desktop displays. HMD systems limit the user's field of view, can become uncomfortable, and often feature a screen that is too dim [17]. Desktop displays are not immersive; in the case of Kolarić et al., the user can see his or her hands in the workspace in front of the monitor.

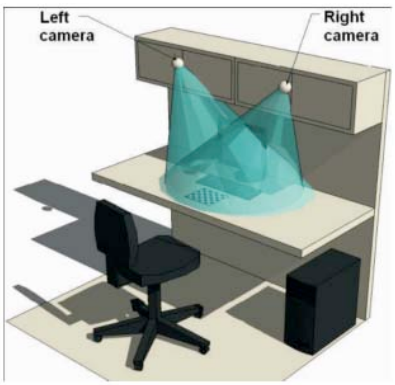

Figure 6: Similar Concept [39]

THE AUGMENTABLE SYSTEM

The proposed system combines a number of well established computer vision techniques with a novel, inexpensive apparatus. This section details the apparatus and algorithms used and how they integrate together.

APPARATUS

A novel element of the AugmenTable system is its ability to provide a near immersive augmented reality experience without requiring the user to wear or hold any devices. The apparatus places a thin-form factor display raised at an angle to face the user (see Figures 1 and 2 above.) The user may sit or stand in front of the display (depending on the height of the table on which it rests) and reach his or her hands underneath and behind the display. A mirror is mounted to the reverse side of the display, reflecting an image of the desktop and the user's hands outwards towards a camera mounted on a tripod. The television, mirror, and additional tracking cameras are all mounted to an adjustable, light weight aluminum frame. The frame is designed to allow adjustment of the height and angle of the television relative to the user. The display and each camera are connected to a PC equipped with multiple core processor(s).

For this working prototype, a Samsung 40" LED TV (UN40B6000VF), three Logitech Webcam Pro 9000 cameras, and a Dell workstation featuring an Intel Xeon X5570 quad-core processor and a Nvidia Quadro FX 4500 graphics card were used. Depending on display size and computer power, a similar functional apparatus could be constructed for less than $4,000.

The AugmenTable provides an immersive experience by simultaneously hiding the user's hands and displaying them in a scene with virtual objects. To be fully immersive, augmented reality should provide visual-spatial, proprioceptive, and haptic cues. Haptic feedback cannot currently be simulated without requiring the user to wear a device, such as in those used in [17] or [13]. Proprioceptive feedback is the mind's self-awareness of the body and is generally a very weak, easily fooled sense - research has shown that human's proprioception is dominated by the visual sense [40]. Visual-spatial cues are the visual phenomenon the brain uses to identify where it is in space relative to other objects. These cues include transparency, occlusion, size, shading gradients, and cross references such as shadows among others [41]. Overall, this system is limited to providing relative size and occlusion cues, masking proprioception, and very simplistic haptic cues when a virtual object is placed against the tabletop surface.

This apparatus proves more immersive than many other current augmented reality experiences. First, the experience of this system is more immersive than a traditional desktop monitor. By hiding the hands from the user and showing a representation of the hands within the virtual world, the user does not have to resolve seeing his or her hands in front of him or herself with also seeing his or her hands in a different location. This advantage may not be largely significant given human ability to map control of the body to manipulation of distant objects, as typified by using steering wheels, game controllers, and laser pointers for example.

More commonly, AR is provided through hand held devices such as mobile phones or tablet computers. These systems do not typically include any part of the user within the augmented reality "window", due to the user having to hold the device in place. In research, the HMD is the most frequently used device for experiencing augmented reality.

The AugmenTable offers both pros and cons compared to these two standards. A mobile phone/computer can provide augmented reality anywhere the user takes the device; the apparatus described here is stationary. An HMD provides a direct angle of view for the user to experience augmented reality; the apparatus described here will most likely display an angle of view slightly different than the user's direct gaze due to the stationary camera. HMDs also provide stereo viewing capability that is currently lacking in the AugmenTable. However, this apparatus does not require the user to carry a

device, provides a large field of view that can eclipse the user's peripheral vision, and does not require the user to wear heavy equipment on his or her head. Finally, although the ultimate goal of the AugmenTable is to enable natural interaction via markerless hand tracking, this initial implementation incorporates a minimal fingertip marker in the form of a colored finger cap (i.e., thimble). This improves the illusion of direct manipulation for the user and reduces the overhead of starting to use and learn the system.

This apparatus provides a good baseline immersive experience for an augmented reality workstation. Improvements are described in the Future Work section that could improve the experience even further.

SOFTWARE LIBRARIES

Most interesting software projects today would not be possible without having powerful libraries to stand upon. The AugmenTable relies on three libraries for a significant number of tasks; each was essential, and a significant effort was made to integrate them together.

First, as mentioned above, is the ARToolkit library [12]. ARToolkit is used here for a number of important initialization steps: determining camera distortion parameters, searching a 2D image for a stored 2D marker pattern, and calculating the inverse camera matrix based on the size and orientation of the detected marker. ARToolkit does not perform all of these steps perfectly, unfortunately. The 2D marker detection can be vulnerable to false positives. In this case, the system requires a recalibration before use. Currently, the system uses ARToolkit version 2.72.1. ARToolkit should be replaced with a more reliable augmented reality library in the future.

Second is the ubiquitous computer vision library OpenCV. OpenCV provides access to the raw camera image feeds, matrix calculation operators, and important 2D image processing algorithms such as color histogram matching, morphology operations, and contour detection. Each of these algorithms is discussed in depth below. The AugmenTable currently uses OpenCV version 2.1.0.

The third and final library used in the creation of this system is OpenSceneGraph. OpenSceneGraph is used to create and manage the 3D scene that comprises the augmented reality environment. It handles all three dimensional models, lights, and events including model intersections necessary for all interactions. OpenSceneGraph 2.7.2 is the current version used in this system.

APPLICATION ARCHITECTURE

To demonstrate the feasibility of the AgumenTable, an application was developed to simulate the assembly of a printed circuit board. Such an application has potential uses in prototype design assessment or training of assembly personnel.

The software architecture of the AugmenTable is depicted in Figure 9. In this initial implementation, the central camera provides video input to ARToolKit to provide tracking of parts via standard fiducial markers. In addition ARToolKit tracks a fiducially marked interaction device in the form of a "pointer". The central camera also provides input to the "Hand Shader" module which incorporates a GPU shader to provide occlusion of virtual objects with physical (e.g., the hand), as described below.

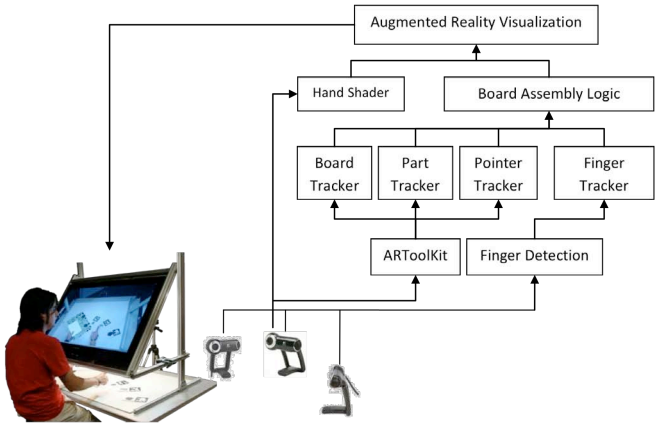

Figure 9. AugmenTable software architecture

All three cameras provide input to the "Finger Detection" module, which incorporates OpenCV to provide tracking for color-marked fingers. Although this function is feasible with fewer than three video inputs, the AugmenTable incorporates three video feeds to enable additional research in markerless finger tracking described in the Future Work section.

The "Board Assembly Logic" module encapsulates the spatial relationship of component parts with the base circuit board and implements a simple proximity "snap-to" function so that as a user brings a part into its approximate position relative to the board, it snaps into its assembled position.

In this application, the user is provided with a physical prop (a piece of white cardboard with fiducial markers) that represents the circuit board. In addition, three smaller fiducially-marked cardboard props are placed on the desktop to represent bins containing different component parts.

In operation, the AugmenTable enables two-handed interaction to simulate the assembly process of the printed circuit board. The user typically grabs the board with one hand, and then selects a part from a part bin, with either the fiducially-marked pointer, or with a color-marked finger, of the other hand. A proximity switch is also implemented to affect part selection. As the finger (or pointer) becomes spatially proximate to a part bin, the virtual representation of the part snaps onto the finger (or pointer), and moves with it. Part assembly is accomplished when the part is brought into the approximate vicinity of its appropriate location on the circuit board, at which time it snaps into assembled position.

Since both hands are active in the assembly process, the AugmenTable provides a very natural interaction metaphor as the user intuitively manipulates the board, relative to their view, to facilitate easy interpretation of the proper location for the part. The following sections describe the highlights of the AugmenTable software application internals including initialization, the hand shader and finger tracker modules.

INITIALIZATION

To efficiently exploit concurrent data from multiple cameras, the process is broken into multiple, parallel threads. This enables the system to function in real time on modern multi-core processors. The process begins with initialization: in addition to typical variable and memory initialization, each camera calculates its position in space prior to starting image processing. Using ARToolkit, each camera searches for a predefined marker in its field of view. Upon locating the marker, its size and orientation are compared to the known marker parameters. This comparison enables ARToolkit to calculate the distance and orientation of the marker compared to the camera in matrix form. The inverse of this matrix results in the camera's position and orientation relative to the marker. Each camera's viewport, projection, and model view matrices are calculated in this way and stored for future reference.

After calibration, the ARToolkit marker is extraneous; the system currently does not attempt to update the camera positions unless the user manually instigates a reset. This is due in part to an attempt to reduce the computational complexity of each frame update, but also because the basic ARToolkit is not capable of the parallel processing necessary to operate across multiple threads. ARToolkitPlus and other subsequent AR libraries have support for parallel processing, so if CPU bandwidth is available it may be possible to update the camera position on the fly, making the system more robust to movement and vibration.

The final initialization step is to store a background image for each camera, with no real objects in the work area. This image is used to facilitate occlusion as described below.

Following the initialization of each camera, the scene (rendered through OpenSceneGraph) is created and the processing threads are executed. A thread is created for each camera to perform image processing in parallel with one additional thread to perform 3D calculations and point tracking.

FINGER TRACKER

Although the ultimate goal of the AgumenTable is to enable completely markerless user interaction, this initial implementation utilized small (red) colored markers on the users index fingers. The finger tracker module implements simple color thresholding to identify finger tips in each camera frame, ray intersection to compute finger tip points in 3D, and a Kalman filter to track them.

Color thresholding is one of the most basic techniques employed in image processing. In most images, individual pixels values range from 0 to 255 in red, green, and blue channels (RGB.) Thresholding is accomplished by finding pixels with the desired common RGB levels within specified RGB ranges. For general applications, such as skin detection, color thresholding can be challenging due to variations in skin tone and lighting. In this application, however, given the AugmenTable's limited workspace with white background, uniform constant lighting and red finger markers, RGB threshold parameters are tuned to easily identify a sufficiently sized cluster of red pixels in each image. This relatively tight color threshold also made additional preprocessing steps, such as background segmentation, unnecessary, and since color thresholding is very fast (O(n) calculations) it runs in three parallel threads; one for each camera. Next, each thread of the finger tracker computes the centroid of the cluster of red pixels to approximate the fingertip in image space.

Once a set of two-dimensional fingertip points have been identified, the next step is to calculate the three dimensional position of the fingertip. The AugmenTable accomplishes this task through ray intersection. Using the camera's matrices computed in the initialization phase, the 2D candidate points are calculated at the near and far plane of the camera's view volume. These two points are the endpoints of a ray segment that projects through the scene. A ray is developed for every candidate point for every camera. The AugmenTable is designed in such a way that the camera view volumes intersect to create the working volume behind the display.

To calculate the fingertip points in 3D, a separate ray intersection thread calculates how close each ray from each camera passes to the other cameras' rays. As shown in Figure 10, the midpoint of the minimum-distance line segment between each ray is computed. The vector algebra of this calculation can be found in [52]. The centroid (spatial average) of these three points is computed as the approximate fingertip location in 3D.

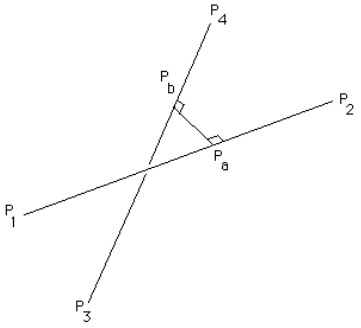

Figure 30: Distance between rays [52]

Most AR+gesture systems track only the hand without attention to fingers. These systems use a variety of established tracking algorithms such as optical flow in [53] and [43], MeanShift and continuously adaptive MeanShift in [54] and [55], the Viola-Jones tracking algorithm in [47] and [29], the KLT tracking algorithm in [51], and [56] and the more recent SIFT/SURF techniques [43] or condensation algorithms [57]. These algorithms, though powerful, are unable to track an arbitrary, changing number of objects – like the number of fingertips visible to a camera.

Fingertip tracking as a topic does not attract the same interest as the broader problem of hand tracking, though many of the same issues apply. Since this system is intended for natural object manipulation, the only features necessary to track are individual fingertips. Hand orientation information is not necessary. Two methods are popular in literature for tracking individual points: Kalman filtering and particle filtering.

A Kalman filter creates a (typically linear) model of a point and its movements [58]. The filter creates a prediction of the point's movement based on the model and is iteratively updated based on the measurement of the point's actual movement. Kalman filters are appropriate when the error in the measurements are Gaussian, but otherwise tend to make

erroneous predictions. This method is useful for tracking a marked finger in stereoscopic environment [59].

Kalman filters are considered "single hypothesis" filters, meaning the filter has only one guess about where the tracked point could be. Multiple hypothesis filters exist, most notably particle filters. Particle filters create recursive Bayesian estimates of particles based on measurements and are suitable for tracking points that may have multiple likely positions at a given time. Particle filters have been used for hand tracking by [26], [27], [60], [61], and [55].

At first glance, it would seem that particle filters are more appropriate for this system because it has to track multiple fingertips through motion that is not linear and unlikely to have Gaussian measurement errors. However, particle filters require significantly more computing power to run. To ensure real-time or near real-time processing speeds and to reduce complexity, this system employed a form of Kalman filtering.

Tracking is currently paired with the ray intersection calculation in an independent thread. As previously described, the tracking system receives 3D points representing fingertips each update. These points may include clusters of false positives around the fingertips. The system does not currently support any interactions of fingers pressed together so if candidate points are within three centimeters apart, they are merged (averaged) into one.

The tracking algorithm maintains a vector of tracked points and velocities. Each iteration, every point is updated with its linear velocity vector. The updated points are then paired with candidate points based on shortest geometric distance. That is, the system determines the closest pair of tracked and current points. The tracked point and velocity are updated based on the new point using a moving average calculation. Thanks to a relatively small number of points and the thread's processing speed, a 20 frame moving average is calculated without noticeable lag. The updated points are then removed from the lists. This process continues until all tracked points or all detected points have been updated.

Tracked points that do not find a candidate point for updating are left as-is and allowed to persist for up to 15 frames without an update. If no update is found at that point, the tracked point is removed from the system. Candidate points that are left over without a corresponding tracked point are added to the vector of tracked points for future iterations.

This system provides acceptable tracking of fingertip points. In parallel, a list of indices to tracked points is maintained such that the main application thread can track individual tracked points, enabling interactions like translation and rotation using the fingertips.

This Kalman-like tracking system is fundamentally similar to a method described by Argyros and Lourakis [62]. Argyros and Lourakis developed a system using adaptive skin histograms and blob tracking which used iteratively updated hand position hypotheses to follow the hands. Their hypotheses in turn were robust against changes in momentum and even occlusion. Further improvements to this system's tracking could be to more fully implement Argyros and Lourakis' statistical tracking methods.

HAND SHADER

Virtual and real objects are mixed in every augmented reality application. In this case, the user can see his or her own hands using the AugmenTable's video see-through display, and visual-spatial cues are the visual phenomenon the brain uses to identify where the hand is in space relative to objects. Thus, occlusion is necessary to know where virtual object are and to touch them.

The hand shader makes possible correct occlusion between real and virtual objects. An image based occlusion algorithm has been developed and implemented as a GPU shader. Since the central camera of the AugmenTable provides the primary visual display for the user, it is the only channel required to accomplish occlusion. The hand shader implements the following four-step occlusion process:

1. ***Background subtraction***. In each frame the image is compared with the calibrated background image that was stored in the initialization phase, and a foreground mask is generated. Every pixel recognized as background will be displayed under the rest of the objects, virtual or real.
2. ***Marker board recognition***. Using the foreground mask, ARToolkit markers are recognized using a color binarization since the ARToolkit fiducial markers are always black and white images. The markers are always covered with their virtual representation, so they are displayed under the virtual objects.
3. ***Real object recognition***. The rest of the foreground, which is not recognized as a marker, is considered a real object. These objects will be always displayed above the virtual ones.

This simple technique works correctly, and it is very fast because it avoids complex image analysis and it is executed in the GPU, which processes multiple pixels at the same time (parallel processing). Despite its computational efficiency, this method produces the correct spatial-visual mix of real and virtual objects, as shown in Figure 11. Without the hand shader (left-hand image), the virtual object is above the user's hands and, since they are covered by the fiducial markers (which are always covered by the virtual part), the hand appears under the virtual part. With the hand shader (in the right-hand image), although the virtual object is under the real hand, it is recognized as a real object and displayed above every virtual object, so the occlusion is correct.

CONCLUSION

Originally, the vision for this project was for a person to reach out a hand and move and inspect a virtual object in a work environment. The large majority of this vision has been realized. A user can walk up to the AugmenTable, reach under and behind the display and manipulate virtual objects and real

objects side by side with only a small colored marker on his fingertip. In addition, the system can recognize hands or fingers to affect occlusion of virtual objects by the user.

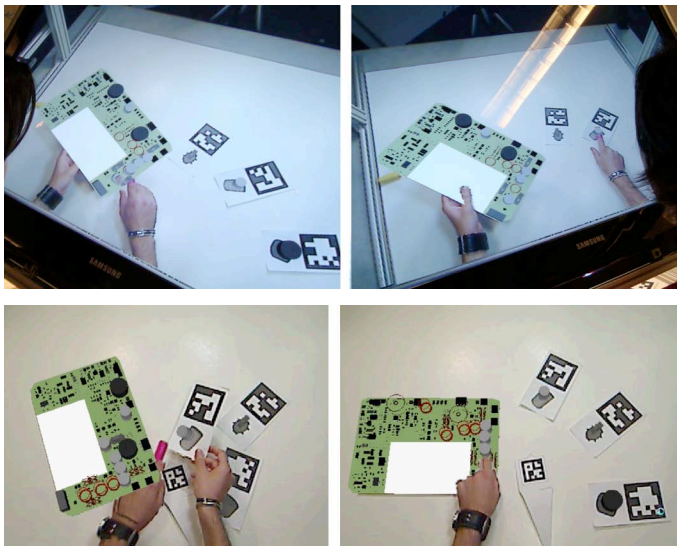

Figure 11: Hand shader occlusion, without occlusion (left); with occlusion (right)

This system exemplifies some novel ideas within the augmented reality research field. The AugmenTable is a unique apparatus that expands the common mobile, hand-held window metaphor into a large-scale desktop system. This scaling of the AR window begins to take on characteristics of the more immersive HMD setup by expanding the view to encompass more of the user's field of vision and allowing the placement of the user's body (in this case hands and arms) within the AR environment. Like an HMD, this setup acts as an intermediary between reality and the user's vision, enabling more rich mixed reality experiences, but without the added steps of donning an uncomfortable head-mounted piece of hardware. This has proven to be an advantage in demonstrating the system's capabilities. The apparatus was set up at a conference alongside a typical HMD system and received noticeably more attention and use.

The AugmenTable also proves successful in realizing a believable mixed reality environment. Through the use of visual display, hand obfuscation, occlusion, and some quasi-haptic feedback (as provided by the tabletop surface,) the system provides a suspension of disbelief about the nature of the virtual objects within the workspace scene. This suspension is not complete. A user still has to use a constructed interaction technique to manipulate virtual objects, but it can be effective.

One final benefit is the low price tag. All of the hardware used is commonly accessible and inexpensive. Custom or expensive components (and the algorithms that rely on them) were purposefully avoided. The largest expense in the project is the multi-core workstation PC. Similarly, all of the software libraries used are open source and freely available. Total hardware costs are less than $4,000 today and the software only had personal time as an expense.

FUTURE WORK

The immediate focus of ongoing research using the AugmenTable is to develop completely markerless hand interaction. The approach under development incorporates a series of well-established image processing techniques including background segmentation, skin detection, and contour evaluation to identify candidate fingertips in each of multiple video feeds. The (approximate) intersection of rays emanating from each camera location through candidate fingertip points in image space provides a 3D location for the fingertip. These points are used to control 3D interaction widgets attached to objects to affect rotation, scaling and other common manipulation tasks.

The intuitive aspects of the prototype applications can be improved upon by the addition and extension of realistic (or at least intuitive) physics. Elements of gravity would add to the immersiveness of the applications and enable both fun and practical interactions. Similarly, giving objects a level of mass or inertia (like that seen in the multitouch swipe gestures of interfaces such as the iPad) can increase the power of gestures without reducing the benefits of direct manipulation. Physics and mass would enable another set of interactions such as pinch, bump, and momentum transfers. More broadly, physics could possibly be extrapolated to creating shadows of objects that would provide an additional depth cue and increase the melding of virtual and real within the workspace. Finally, physics provides an overall expectation of interaction. Users are accustomed to the physical world where objects behave in a reliable manner due to the laws of physics. With a software physics engine, a similar expectation is created in a virtual environment. This expectation allows users to more easily extrapolate real actions to virtual actions. Physics is therefore a significant step to opening an augmented reality to arbitrary object manipulation without intermediary widgets.

This implementation of the AugmenTable apparatus also could benefit from a number of additional advanced technologies. For instance, 3D displays are coming onto the market in 2010. The thin display here could be replaced with a 3D capable display and provide stereo perception to improve the immersiveness. Another possible display change would be swapping the simple display with a multitouch display. This could enable a mixture of 2D and 3D control of virtual objects and interfaces. The hand shader could be enhanced substantially by incorporating a low cost depth camera such as Microsoft's Kincet. The depth information provided could be incorporated directly into the GPU shader to improve accuracy.

One possibility would be to add a forward facing camera that provides face tracking of the user. Face tracking can enable changing the perspective of the display in order to provide correct occlusion of objects relative to the user's perspective. This creates a much greater three-dimensionality effect than stereo display alone, and is much cheaper than the nascent 3D monitor technology.

Finally, this research would benefit from a user study of the various interaction techniques developed to compare their intuitiveness or learn-ability to other methods.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Rockwell Collins, Inc. for financial support of this research. In particular, the technical creativity of Rockwell colleagues James Lorenz, Ryan Wheeler and Daniel Turner is greatly appreciated. The authors also acknowledge the work of Erik Steindecker (Technical University of Dresden) and Arnaud Martin (Ecole Nationale Supérieure d'Arts et Métiers), both of whom contributed to developing and evaluating earlier prototypes of the AugmenTable.

REFERENCES

[1] B. Shneiderman, "Direct manipulation," *Proceedings of the joint conference on Easier and more productive use of computer systems. (Part - II) Human interface and the user interface*, 1981, p. 143.

[2] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, and G. Rigoll, "Static and dynamic hand-gesture recognition for augmented reality applications," *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, 2007, p. 728–737.

[3] A. Eliott, "10 Amazing Augmented Reality iPhone Apps," *http://mashable.com/2009/12/05/augmented-reality-iphone/*, 2010.

[4] O. Inbar, R. Nir, and T.K. Carpenter, "Games Alfresco," *http://gamesalfresco.com/*, 2010.

[5] Google, "Google Goggles," *http://www.google.com/mobile/goggles/#landmark*, 2010.

[6] P. Mistry and P. Maes, "Sixth sense: integrating information with the real world," *http://www.pranavmistry.com/projects/sixthsense/*, 2010.

[7] C. Harrison, D. Tan, and D. Morris, "Skinput: Appropriating the Body as an Input Surface," *Proceedings of the 28th Annual SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, Georgia: 2010.

[8] D. Dumas and Wired.com, "CES 2010: Hands-On With Transparent Display of the Future," *http://www.wired.com/video/ces-2010-hands-on-with-transparent-display-of-the-future/60826805001*, 2010.

[9] S. Kim and A.K. Dey, "AR interfacing with prototype 3D applications based on user-centered interactivity," *Computer-Aided Design*, vol. 42, 2010, pp. 373-386.

[10] G. Bleser, Y. Pastarmov, and D. Stricker, "Real-time 3d camera tracking for industrial augmented reality applications," *Journal of WSCG*, 2005, p. 47–54.

[11] C. von Hardenberg and F. Bérard, "Bare-hand human-computer interaction," *Proceedings of the 2001 workshop on Perceptive user interfaces*, New York, New York, USA: ACM New York, NY, USA, 2001, p. 1–8.

[12] P. Lamb, "ARToolkit," *http://www.hitl.washington.edu/artoolkit/*, 2007.

[13] V. Buchmann, "FingARtips – Gesture Based Direct Manipulation in Augmented Reality," *Virtual Reality*, vol. 1, 2004, pp. 212-221.

[14] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top AR environment," *IEEE and ACM International Symposium on Augmented Reality, 2000 (ISAR 2000). Proceedings*, 2000, p. 111–119.

[15] D. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Computer Graphics and Applications*, 1994.

[16] C. Keskin, A. Erkan, and L. Akarun, "Real time hand tracking and 3D gesture recognition for interactive interfaces using HMM," *ICANN/ICONIPP*, 2003, p. 26–29.

[17] S. Walairacht, K. Yamada, S. Hasegawa, Y. Koike, and M. Sato, "4+ 4 fingers manipulating virtual objects in mixed-reality environment," *Presence: Teleoperators \and Virtual Environments*, vol. 11, 2002, p. 134–143.

[18] J. Rehg and T. Kanade, "DigitEyes: vision-based hand tracking for human-computer interaction," *Proceedings of 1994 IEEE Workshop on Motion of Non-rigid and Articulated Objects*, 1994, pp. 16-22.

[19] C. Nölker and H. Ritter, "Detection of fingertips in human hand movement sequences," *Gesture and Sign Language in Human-Computer Interaction*, Springer, 1998, p. 209–218.

[20] K. Abe, H. Saito, and S. Ozawa, "Virtual 3-D interface system via hand motion recognition from two cameras," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 32, 2002, pp. 536-540.

[21] W. Chen, R. Fujiki, D. Arita, and R. Taniguchi, "Real-time 3D Hand Shape Estimation based on Image Feature Analysis and Inverse Kinematics," *14th International Conference on Image Analysis and Processing (ICIAP 2007)*, 2007, pp. 247-252.

[22] K. Oka, Y. Sato, and H. Koike, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems," *Proceedings of the fifth IEEE international conference on automatic face and gesture recognition*, IEEE Computer Society Washington, DC, USA, 2002, p. 429.

[23] Y. Sato, Y. Kobayashi, and H. Koike, "Fast tracking of hands and fingertips in infrared images for augmented desk interface," *International conference on automatic face and gesture recognition*, Grenoble, France: 2000, pp. 462-467.

[24] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, 2007, pp. 52-73.

[25] L. Bonansea, "3D Hand gesture recognition using a ZCam and an SVM-SMO classifier," *Journal of empirical research on human research ethics : JERHRE*, vol. 5, 2010.

[26] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, 2002, pp. 423-428.

[27] T. Gumpp, P. Azad, K. Welke, E. Oztop, R. Dillmann, and G. Cheng, "Unconstrained Real-time Markerless Hand Tracking for Humanoid Interaction," *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 88-93.

[28] S. Kang, M. Nam, and P. Rhee, "Color Based Hand and Finger Detection Technology for User Interaction," *Convergence and Hybrid Information Technology, 2008. ICHIT'08. International Conference on*, 2008, p. 229–236.

[29] M. Kolsch and M. Turk, "Robust hand detection," *Proc. of the Sixth IEEE Int. Conf. on Automatic Face*, vol. 17, 2004, pp. 614-619.

[30] C. Malerczyk and G. Darmstadt, "Dynamic Gestural Interaction with Immersive Environments," *Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2008.

[31] C. Hand, "A survey of 3D interaction techniques," *Computer graphics forum*, vol. 16, 1997, pp. 269-281.

[32] J. Segen and S. Kumar, "Gesture vr: vision-based 3d hand interace for spatial interaction," *Proceedings of the sixth ACM international conference on Multimedia*, ACM New York, NY, USA, 1998, p. 455–464.

[33] K. Oka, Y. Sato, and H. Koike, "Real-time fingertip tracking and gesture recognition," *IEEE Computer Graphics and Applications*, vol. 22, 2002, pp. 64-71.

[34] P. Song, S. Winkler, S. Gilani, and Z. Zhou, "Vision-based projected tabletop interface for finger interactions," *Lecture Notes in Computer Science*, vol. 4796, 2007, p. 49.

[35] T. Lee and T. Höllerer, "Hybrid Feature Tracking and User Interaction for Markerless Augmented Reality," *2008 IEEE Virtual Reality Conference*, 2008, pp. 145-152.

[36] T. Lee and T. Höllerer, "Handy AR: Markerless inspection of augmented reality objects using fingertip tracking," *International Symposium on Wearable Computers*, Citeseer, 2007, pp. 83-90.

[37] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *IEEE transactions on visualization and computer graphics*, vol. 12, 2006, pp. 615-28.

[38] P. Song, H. Yu, and S. Winkler, "Vision-based 3D finger interactions for mixed reality games with physics simulation," *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, ACM, 2008, p. 7.

[39] S. Kolarić, A. Raposo, and M. Gattass, "Direct 3D Manipulation Using Vision-Based Recognition of Uninstrumented Hands," *Symposium of Virtual and Augmented Reality*, 2008, pp. 212-220.

[40] M.S. Graziano, "Where is my arm? The relative role of vision and proprioception in the neuronal representation of limb position," *Proceedings of the National Academy of Sciences of teh United States of America*, vol. 96, 1999, pp. 10418-10421.

[41] C. Furmanski, R. Azuma, M. Daily, and H.R. Laboratories, "Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information," *Symposium A Quarterly Journal In Modern Foreign Literatures*, 2002.

[51] Y. Pang, M.L. Yuan, A.Y. Nee, S.K. Ong, and K. Youcef-toumi, "A Markerless Registration Method for Augmented Reality based on Affine Properties," *Proceedings of the 7th Australian User Interface Conference*, Hobart, Australia: 2006, pp. 24-32.

[52] P. Bourke, "The Shortest Line Between Two Lines in 3D," *http://local.wasp.uwa.edu.au/~pbourke/geometry/lineline3d/*, 1998.

[53] F. Dadgostar and a. Sarrafzadeh, "An adaptive real-time skin detector based on Hue thresholding: A comparison on two motion tracking methods," *Pattern Recognition Letters*, vol. 27, 2006, pp. 1342-1352.

[54] T. Kurata, T. Okuma, M. Kourogi, and K. Sakaue, "The Hand Mouse: GMM hand-color classification and mean shift tracking," *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 2009, pp. 119-124.

[55] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, 2007, pp. 1958-1970.

[56] M. Kolsch and M. Turk, "Fast 2d hand tracking with flocks of features and multi-cue integration," *CVPRW'04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04*, Citeseer, 2004, p. 158.

[57] E. Koller-Meier and F. Ade, "Tracking multiple objects using the condensation algorithm," *Robotics and Autonomous Systems*, 2001, pp. 1-18.

[58] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, 1960, pp. 35-45.

[59] K. Dorfmuller-Ulhaas and D. Schmalstieg, "Finger tracking for interaction in augmented environments," *Proceedings IEEE and ACM International Symposium on Augmented Reality*, IEEE Comput. Soc, 2001, pp. 55-64.

[60] I. Laptev and T. Lindeberg, "Tracking of Multi-state Hand Models Using Particle Filtering and a Hierarchy of Multi-scale Image Features," *Scale-Space and Morphology in Computer Vision*, Berlin: Springer Berlin/ Heidelberg, 2001, pp. 63-74.

[61] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," *Lecture Notes in Computer Science*, vol. 1843, 2000, p. 3–19.

[62] A. Argyros and M. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera," *Lecture Notes in Computer Science*, 2004, p. 368–379.

