RC22495 (W0205-143) May 17, 2002
Computer Science

IBM Research Report

Ubiquitous Interactive Graphics

Claudio S. Pinhanez, Frederik C. Kjeldsen, Anthony Levas,
Gopal S. Pingali, Mark E. Podlaseck, Paul B. Chou

IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Ubiquitous Interactive Graphics

Claudio Pinhanez, Rick Kjeldsen, Gopal Pingali, Anthony Levas, Mark Podlaseck, Paul Chou
IBM Research, T.J. Watson
pinhanez@us.ibm.com

Abstract

This paper presents a device that creates ubiquitous interactive graphics in a real world environment without the need for pre-wiring surfaces or requiring users to carry or wear any special devices. Called the *Everywhere Displays projector*, or *ED-projector*, this device employs *steerability* as the key principle to achieve the promise of ubiquitous computing and augmented reality. The ED-projector uses an LCD projector with a steerable mirror to project graphics onto any pre-calibrated surface, while warping the projected image to correct for oblique distortion. Three different methods to determine this warping function are given. To sense user interaction with the projected image, the ED-projector employs a pan/tilt/zoom camera and a motion-based computer vision system to track the user's hand and touch-like gestures. We propose a system software structure that ties together steering, display, and sensing. Three implemented applications demonstrate the notion of ubiquitously "painting" the real world with interactive graphics, in the context of a futuristic office, an augmented assembly task, and a ubiquitous computer game. The observation of hundreds of users performing the assembly task suggests that the traditional desktop paradigm is inadequate for ubiquitous interaction and that new conceptual widgets and interaction paradigms must be developed.

Keywords: ubiquitous computing; augmented reality; gesture recognition; interactive projections; ubiquitous games.

1 INTRODUCTION

The extraordinary success of computer graphics technology in recent years has been responsible for the creation of amazing new worlds in movies, interactive computer games, and virtual reality (VR) environments. However, this very success poses the question of when the real world, where we live and work, is going to be as rich with imagery and information as movies and games today.

This paper describes technology that can be used to ubiquitously augment, correct, and enhance reality. Although the ultimate goal is to create interactive pixels anywhere in a space as seamlessly as it happens in movies (thanks to CG special effects), this particular work focus on how to "paint" interactive graphics on the surfaces of an environment without requiring any kind of fixed display devices, special wiring of surfaces or the use of encumbering headsets.

The system described here is based on a new device, the *Everywhere Displays projector* (*ED-projector*), composed of a steerable projection system and a pan/tilt camera. A prototype of the projection system has been built using a pan/tilt mirror that directs imagery generated by a standard LCD projector. To capture user interaction with the projected graphics, the images gathered by the camera are processed using computer vision techniques. In the current prototype, it is possible to determine the user's hand position in referennce to a surface and to detect simple, touch-like gestures, effectively mapping the user's hand gestures into mouse-like events.

For instance, we have developed a computer game, called "Fro…og!", where a computer generated image of a frog initially appears "sitting" on the side of a file cabinet (see Figure 1). If a player tries to catch it, the vision system detects the player's hand approaching and then shows an image of frog jumping (see Figure 1). This is followed by moving the image of the frog to another surface, for instance the top of a table by appropriately steering the mirror. All this happens in an environment under normal lighting conditions, ultimately creating the illusion that the frog jumps around the space to avoid being caught by the player.

Our goals, however, go beyond transforming the real world into a magical entertaining place. In particular, we envision the creation of seamless interfaces to devices on surface of walls and everyday objects such as tables, chairs, couches, doors, cabinets, and carpets. Currently, only very "special" surfaces such as TVs, monitors, and projected screens support the creation of computer graphics, and usually require external devices such as mice and joysticks to make them interactive. In our paradigm, any surface is regarded as generic support for interactive graphics. By projecting imagery and detecting user interaction with a camera, it is possible to transform every surface into a virtual touch-screen.

These ideas have been explored in three application prototypes

Figure 1. Trying to catch a projected frog.

described in the paper. The first in an attentive/interactive office space.The second is in the context of augmenting the environment of an assembly task by projecting interactive graphics directly onto the surfaces of the objects involved. Ultimately we aim to realize Mark Weiser's paradigm of ubiquitous computing (see [1]) without having to change the fabric of the world by embedding touch-screens on every surface. The thir application is the ubiquitous computer game "Fro…og!".

This paper starts by comparing our approach and solutions with previous work, followed by a description of the technology used in the prototypes we have built. Although initial ideas on the ED-projector have been reported in a previous publication [2], this paper presents the device comprehensively including mathematical formulation, computer vision based interaction, and three implemented applications. Specifically, this paper describes the mathematical formulation of the method for correcting oblique distortion and presents two new calibration methods for the first time. Further, in this paper we describe an integrated, working computer vision system that we implemented to track hands and detect touches, as well as a way to structure the interface to application software. While [2]ellaborates only on conceptual mock-ups, this paper describes three implemented applications using the ED-projector, followed by a discussion of how novice users — in one case, more than 600 of them — have experienced and interacted with the application prototype. We conclude by examining other applications that can take special advantage of the ED-projector and discuss the challenges these new interfaces and applications create for HCI research.

2 RELATED WORK

In his seminal paper that defined the field of ubiquitous computing, Weiser [1] envisioned computers embedded everywhere — ubiquitous and transparent. In his view, interaction with computers would be accomplished both through direct mechanisms linking sensors and actuators and by the widespread presence of cheap interactive screens (referred as "pads") embedded on the surfaces of walls, furniture, and objects. Since the publication of that paper, different devices and methods to implement the functionality envisioned for Weiser's pads have been proposed. In general, those devices and methods can be divided into four categories: head-mounted, portable, embedded, and projection-based.

A good survey of the current research on head-mounted devices and their uses in ubiquitous computing is provided by Barfield and Caudell [3], describing both wearable computers and the more traditional approach of off-body computer power. We see two main challenges in this approach First, in order to really connect to the physical world, it is necessary to precisely register the position and attitude of the head of the user with the environment in real time. An examination of the state-of-art of the research in the area shows that this is currently achievable only in very constrained environments (see, for example, Raskar et al. [4]).

The second challenge with head-mounted methods for ubiquitous displays is that they do not work well in social contexts and collaborative tasks. If multiple users are looking at the same surface (for instance, a whiteboard), it is necessary to render exactly the same graphics on each user's head-mounted display simultaneously. This creates considerable difficulties in terms of accomodating different resolutions, rendering speeds, etc. of different brands and models of head-mounted displays. In practice, it is very hard to guarantee that all participants are seeing exactly the same images.

The second method being studied to create ubiquitous displays utilizes portable devices such as laptops, PDAs, and mobile phones. Like head-mounted displays, portable devices do not work well in social and collaboration tasks (except in very unusual conditions such as in multi-player games [4]), and are a nuisance to carry and power.

The third method involves embedding screens or similar interaction devices into the objects themselves, as proposed initially by Weiser [1] and more recently by Ishii [5]. However, even if screens became extremely cheap, this approach requires a widespread change in the way everyday objects are manufactured and installed. Power and network access (or wireless bandwidth) has to be provided to even the simplest object in a home or workplace. In other words, this approach requires a drastic change of the very fabric of the real world.

Our approach follows the fourth method of creating ubiquitous displays that is based on projection devices. Interestingly, the use of projectors to change and augment reality was pioneered by media artists such as Tony Oursler [6] and Michael Naimark [7]. The use of projectors to access computers have been initially proposed by Bolt [8] and Wellner [9]. Recently, more radical applications of projectors have been proposed by Morishima et al. [10] that enhance the face of an actor, and by Raskar et al. [11] to change the color and texture of real objects such as vases and architectural models. Although our system does not currently track moving surfaces as in the former example, or project onto 3D objects as the latter, it can clearly incorporate both features in future versions.

Deviceless interaction with projections was pioneered by Krueger [12], using camera-based systems to detect the users' hand and body. The fact that most vision techniques put strict constraints on lighting conditions and on the background surfaces has restricted the use of these systems to entertainment applications such as Keays and McNeil's *metaField* [13]. To avoid lighting constraints, Omojola et al. [14], Ju [15], and the RED project at Xerox Parc [16], among others, embedded sensors on the surface where the images are projected. As in the embedded approach described before, the widespread adoption of sensor-loaded surfaces requires a dramatic change in the objects of the real world. Underkoffler and Ishii [17] and Rekimoto [18] used visual tags to detect the position and the identity of the objects on a surface and to overlay the interaction based on these attributes. To avoid the need of tagging all objects in an environment as in those two cases, a possible option is the use of object recognition techniques (such as those described by Schiele and Crowley [19]). However, those techniques currently work only in very constrained situations. We believe, though, that incorporating object recognition capabilities into our system is an important direction to follow, and we plan to do so forspecific applications.

Vision-based gesture recognition has received a great deal of attention in recent years [20], but only a little of that work has attended to the problems of gesture recognition in the context of a projected display. Because the projected image drastically changes the appearance of the hand, most approaches to hand detection are based on either background subtraction or motion (frame-frame differencing). Our experience agrees with that of von-Hardenberg and Berard [21] in that background subtraction does not give reliable results in the presence of strong projected images.

Gesture-based interaction methods tend to fall into one of two catagories: either variations of a point-and-click paradigm [21, 22] or application-specific pose or motion gestures [23]. Both these approaches have disadvantages. The point-and-click approach is often not well suited to hand pointing due to limited resolution and the lack of a natural "click". On the other hand, a proliferation of application-specific gestures will make for a complex gestural interface that a user will have to learn and execute largely without feedback, as discussed by Kjeldsen in [24].

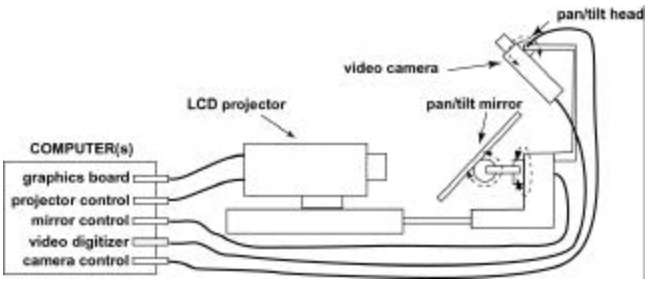

Figure 2. Diagram of the components of the ED-projector.

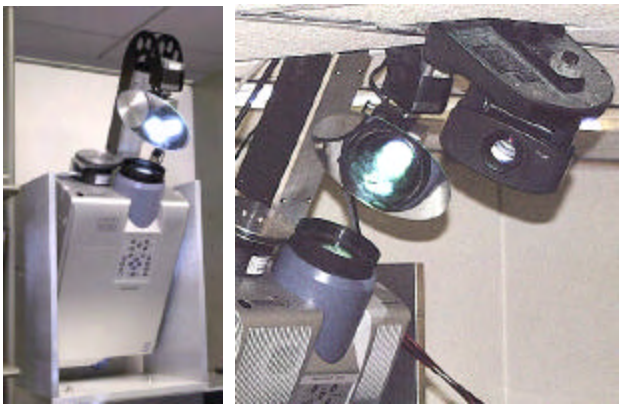

Figure 3. Prototype of the ED-projector: a) ED-projector without the camera mounted; b) a detail of a complete prototype, including the pan/tilt camera.

A major difference between our approach and the cited work on the use of projection-based devices to create ubiquitous interactive displays is *steerability*. By directing the projection and vision subsystems to a relevant position, it is possible to create new "interactive" surfaces in a space without any hardware installation, addition, or change. Unlike Raskar et al. [25], who use mirrors to increase the resolution of multiple-projector installations, we steer the image from a projector to realize interactive displays everywhere in the workspace. By doing so, we opportunistically use the surfaces of the objects present in the environment. We believe that steerability is a key feature that creates a real alternative to realize the promises of ubiquitous computing. However, it has constraints and technical challenges of its own, such as obstruction, oblique distortion, and surface texture interference. These issues are discussed in the next section where solutions to some of these problems are proposed.

3 THE ED-PROJECTOR

Our approach to ubiquitous interactive graphics uses a device — the *Everywhere Displays projector* (the *ED-projector*) — that can steer graphic displays onto any surface and can also sense user interaction with these steered graphical displays. The ED-projector is composed of an LCD projector, a computer-controlled pan/tilt mirror, and a pan/tilt/zoom camera. The projector is connected to the display output of a host computer that also controls the mirror and the camera (see Figure 2). The video output of the camera is sent to a digitizer board in the host computer where the imagery is processed, as described later.

Figure 3 shows a prototype of the ED-projector built with off-the-shelf components — a rotating mirror used in theatrical/discotheque lighting, a standard LCD projector, and a pan/tilt camera. The projector's light can be directed anywhere within a cone defined by approximately 60 degrees in the vertical axis and 230 degrees in the horizontal axis. The camera has a maximum vertical viewing range of 88 degrees and a horizontal range of 249 degrees, and is thus able to view the projected displays as they are steered around.

In general, a projector needs to be able to project a white pattern approximately 10 times brighter than its surroundings to create the illusion of contrast. Our current prototype employs a 3000 lumen projector, which we have found to be sufficiently bright in typical home and office lighting conditions to provide good contrast in the projected displays. Of course, the perceived brightness and constrast is heavily dependent on the color and material of the projected surface. Although our best results involve projection on white surfaces, we have also obtained good results projecting on carpets, black objects, and even translucent backlit surfaces. However, highly textured surfaces such as natural wood and surfaces with high reflectance are, in general, less suited for projected displays.

The three main issues with the ED-projector are 1) correction of the projected image to compensate for distortion due to oblique projection and surface characteristics; 2) sensing the user interactions with the projected displays; and 3) a system software structure that enables applications to easily define interactive display surfaces. These issues are discussed in detail in the rest of this section.

3.1 Correcting the Projected Image

Standard LCD projectors are designed to project light in a direction orthogonal to the projected surface. In the ED-projector, the pan/tilt mirror can deflect the light to multiple surfaces in the room, but most of the times the projected surface is oblique to the direction of projection. The result is a distorted image.

To correct the oblique projection distortion, we pre-warp the image to be projected to compensate for the distortion. It is always possible to correct the projected image, as long as the projected surface does not have a geometry that occludes the cone of projection (see [2] for details). However, the process of warping the image normally uses only a fraction of the projector's image plane, causing loss of resolution. We will discuss this issue later; first, we address the problem of determining the warping function that corrects the oblique distortion.

Although there are many different ways to determine and implement the warping function, it is important to use a method that allows fast computation. We choose to implement it using 3D computer graphics hardware to texture map the images to be displayed onto a mesh in 3D space. If the mesh is appropriately warped and/or positioned in 3D space and the virtual camera correctly positioned, the projection of the resulting rendered image is free of distortion.

The basic structure of the problem, in the case of planar surfaces, is illustrated in Figure 4. We start by defining the *application plane*, a coordinate system where images to be projected are positioned within the area of a rectangular mesh *M*, and the *projector image plane*, the coordinate system associated with the LCD matrix of the projector. We also define a *projected surface* to be the real world surface on to which images are projected. In addition, we have the camera image plane, the coordinate system associated with the CCD matrix of the camera.

The goal is to define a transformation *H* from the application plane into the projector image plane so the transformation that maps the support mesh *M* into the projected surface is the identity *I*, and, therefore, appears free of distortion from an observer orthogonal to the projected surface. Since the process of projecting the image can be modeled as a projective transformation *P* (as

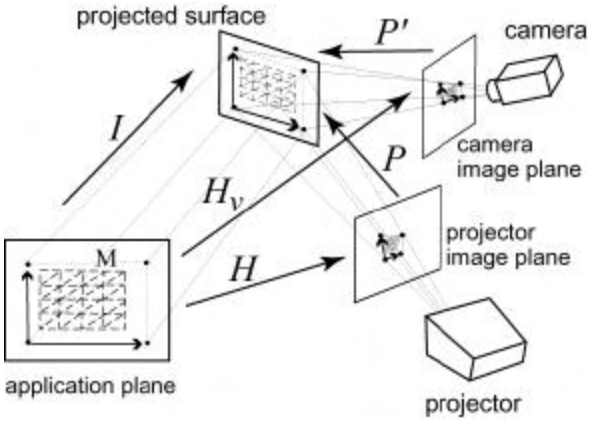

Figure 4. The transformations between the application plane, the projector's image plane, the camera's image plane and the projected surface.

observed in [26, 27]), we can represent the relationships among the planes as matrices in planar homogeneous coordinates, $I = PH$. Since H and P are projective transformations and therefore invertible, we obtain:

$$H = P^{-1}$$

Similarly if P' is the projective transformation between the camera image plane and the projected surface, and H_v is the transformation from the application plane to the camera image plane, we have $I = P'H_v$. Although there is no easy direct way to compute the projection function P, we developed three different methods to determine the actual value of H for a given surface. Of course, H and H_v are different for each planar surface in an environment.

1st method: Using an Equivalent 3D Virtual CG World

This method, first described in [2], is based on the fact that, geometrically, projection is the inverse of camera viewing. So, it is possible to model the projection by simulating the camera view by rotating, scaling, and viewing the support mesh M in 3D space.

Given the n mesh points in 2D projective space, $m_i = [m_i^x \; m_i^y \; 1]$, $1 \le i \le n$, let M be the matrix of all n points, $M = [m_1^T m_2^T \ldots m_n^T]$. This mesh can be "positioned" in the XY plane of the 3D space by multiplying it by the matrix

$$K = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad KM = \begin{bmatrix} m_1^x & m_2^x & \ldots & m_n^x \\ m_1^y & m_2^y & \ldots & m_n^y \\ 0 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 1 \end{bmatrix}$$

We can then simulate the proces of projecting the mesh as the inverse of the process of "viewing" the appropriately positioned mesh in 3D space. That is,

$$P^{-1}M = P_f R_x R_y S_x S_y T_x T_y R_z K M$$

where R_x, R_y, R_z are rotation matrixes in 3D homogeneous coordinates, S_x, S_y are scale matrixes, T_x, T_y are translation matrixes, and P_f is the 3×4 projective matrix with focus point f. Since $H = P^{-1}$, we obtain

$$H = P_f R_x R_y S_x S_y T_x T_y R_z K$$

Notice that all the matrices that compose H, with the exception of K, correspond to standard projection parameters of computer graphics boards. Therefore, to implement this method it is only necessary to determine these individual parameters, load them into the hardware, and command the graphics board to render the scene.

To determine the projection parameters for a given surface we start by texture mapping a calibration pattern to the mesh M. Starting off with default values for the projection parameters that render the mesh on the center of the projector image plane without warping (i.e., $H \approx I$), we iteratively change each of the projection parameters until the projection of the pattern on the projected surface is observed to be identical to the original mesh. We then record these parameters and reload them whenever the ED-projector is aimed at that surface. Although conceptually simple, this iterative process of determining the right projection parameters is typically very time consuming.

2nd method: Manual Determination of Corresponding Points

This and the following method are based on the fact that projective transformations can be completely determined if the corresponding coordinates of four points in each 2D projective space are known (as shown by Faugeras [28]).

In this method, four points $a_i = [a_i^x \; a_i^y \; 1]$, $i = 1,2,3,4$, are directly rendered on the projector image plane. We then iteratively move these points over the projector image space with a pointing device such as a mouse until their projections overlap the location of four known points on the projected surface, $b_i = [b_i^x \; b_i^y \; 1]$, $i = 1,2,3,4$. For greater accuracy, we can choose these four points to be the corners of the mesh M.

It is easy to see that if the projected points are aligned with the projected surface points then $PA = B$, where $A = [a_1^T \; a_2^T \; a_3^T \; a_4^T]$ and $B = [b_1^T \; b_2^T \; b_3^T \; b_4^T]$. By computing the pseudo-inverse,

$$PA = B \rightarrow PAA^T = BA^T \rightarrow P = BA^T \left(AA^T\right)^{-1}$$

Since $H = P^{-1}$, we obtain $H = AA^T (BA^T)^{-1}$. To implement this procedure, we simply construct a warped mesh M' in 3D projective space by computing $M' = KHM$, texture map the image to be projected on it, and use the graphics engine to render it in real time.

As we see, the calibration procedure of this method is quite simple and fast, requiring only a few minutes for the manual alignment of the four projected points. Although similar to the method proposed by [29], here the use of the 3D mesh M and the computer graphics hardware yields much faster rendering speeds.

3rd method: Using a Camera and a Paper Pattern

In this method, we employ the ED-projector camera so we can avoid any manual calibration procedure. In [30], Yang et al. have proposed a method to automatically determine the warping function using a projector/camera pair. Unfortunately, their approach requires precise 3D calibration of the relative attitude of the camera to the projector and, in practice, a considerable baseline distance between them. In particular, it does not work with the ED-projector that aligns the projector and the camera axis as much as possible to avoid user's obstruction of the camera's view.

Instead, we propose here a method that employs a paper pattern positioned on the surface to be calibrated so it is aligned with the desired coordinate system. No relative 3D or 2D calibration is necessary, although we assume that the camera has been positioned in such a way that it has a complete view of the projected surface. Under these conditions, we can define a projective transformation P' between the camera image and the projected surface (see Figure 4). This transformation can be determined automatically by processing an image of the pattern and extracting the camera image coordinates of the corners, $c_i = [c_i^x \; c_i^y \; 1]$, $i = 1,2,3,4$. Assuming the coordinates of the

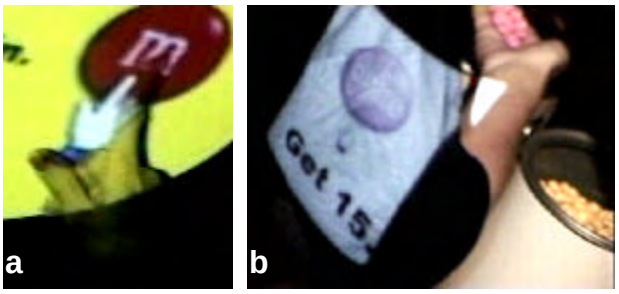

Figure 5. Difficulties faced by the vision system: a) the user's hand is partially hidden by the strong projection; b) the interactive surface is occluded by the user's shoulder as he moves to gather objects from inside a bucket.

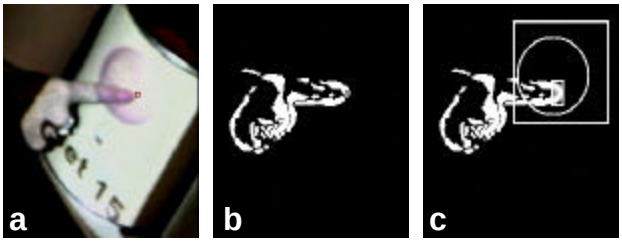

Figure 6. Detecting a touch: a) camera view of an interaction with a bucket; b) image difference data; c) overlay of search region (square), button active area (circle), and the fingertip template shown at the pointing location.

pattern corners to be identical to the mesh M in application space, $b_i = [b_i^x \; b_i^y \; 1]$, $i = 1,2,3,4$, we obtain $P'C = B$, and computing the pseudo-inverse,

$$P' = BC^T \left(CC^T\right)^{-1}$$

To complete the computation of the transformation H, it is necessary to determine the correspondence between the projector and the camera image planes. This can be done by projecting a pattern with four non-colinear points rendered on the projector image plane $d_i = [d_i^x \; d_i^y \; 1]$, $i = 1,2,3,4$, and obtaining four points on the projected surface, $e_i = [e_i^x \; e_i^y \; 1]$, $i = 1,2,3,4$. Then, by image processing the image of the pattern, it is possible to determining the coordinates of its four corners on the camera image plane, $f_i = [f_i^x \; f_i^y \; 1]$, $i = 1,2,3,4$.

Since $E = [e_1^T \; e_2^T \; e_3^T \; e_4^T]$ corresponds both to the projection of $D = [d_1^T \; d_2^T \; d_3^T \; d_4^T]$ by the projector, $PD = E$, and to the captured image of the pattern, $P'F = E$, we obtain $PD = P'F$. Computing the pseudo-inverse, we obtain $P = P'FD^T(DD^T)^{-1}$ and therefore

$$H = DD^T \left(P'FD^T\right)^{-1}$$

As in the previous method, this transformation is implemented by computing a warped mesh $M' = KHM$.

One problem with all these three distortion-correcting methods is that they project displays with resolution lower than the projector's resolution. The distortion-correction process normally fits an irregular quadrangle into the 4:3 viewing area of typical displays. A considerable amount of display area, and thus resolution, is lost in the process. We have employed standard 1024x768 XVGA projectors in our prototypes. Due to the loss of display area created by the distortion correction process, the final display resolution has approximately corresponded to VGA, i.e., 640x480 pixels.

3.2 Sensing Interaction on Any Surface

To support the ability to interact on any surface we use a pan/tilt camera feeding custom image processing and gesture recognition engines. This process has been detailed in another publication [31], but will be outlined here for completeness.

Image processing in this domain is both simplified and made more difficult by the strong light of the projected image. It is simplified in the sense that the lighting environment is very consistent because the dynamic range of the projection swamps all but major changes in ambient light. It is made more difficult because the appearance of the user's hand can be severely distorted as it passes through the projected image (see Figure 5.a).

Although parts of the hand can be hard to detect in a static image, by looking at the differences between adjacent images in the video stream, we can still detect the shape of a moving object as shown in Figure 6.a and b.The use of motion for segmentation instead of background subtraction techniques (such as [32]) also makes it easier to accommodate the variability in the projected image caused by the texture of the surface it is projected on. This motion mask forms the basis of subsequent image understanding.

To provide interaction in a flexible and consistent manner we use the notion of *interface widgets*. Each widget generates computer or application events in response to a particular user motion in a region of the image. We currently use three types of widgets: *buttons* that respond to touch-like motions; *sliders*, 1-dimensional tracking regions, and *touch pads*, which are 2-dimensional tracking regions.

These three widget types rely on tracking the user's fingertip. Convolution template matching is used to find fingertip candidates in the motion mask image described above. The candidates are evaluated using spatial clustering and domain specific heuristics, such as the distance from the user, to determine the best estimate of whether there is a fingertip in the image and where it is currently pointing. The trajectory created by this point over a sequence of video frames is examined by each widget to determine when it should trigger an event.

Touch pad widgets smooth the fingertip trajectory and map it into the touch pad coordinate system. Motion events are only generated when the fingertip is within the touch pad's borders. Sliders are simply a 1-dimensional version of touch pads.

Buttons, however, are more complex widgets. What buttons attempt to detect is when a surface in the environment is touched. Inferring the distance of the fingertip to the surface using the shadow proved to be difficult and unreliable. Fortunately, people tend to interact with real-life buttons by moving in a characteristic forward/pause/backward pattern, and this seems to carry over well to projected buttons. Our implementation of buttons takes advantage of this movementpattern.

To determine when a user has touched a button we look for a roughly linear sequence of fingertip locations that persists for 250ms or longer, with speed within a set range, that either ends or begins with a pause (i.e. the fingertip template did not find a match in the motion mask). A button generates its event when the pause in such a trajectory occurs within it (see Figure 6.c). In other words, we are looking for the user's fingertip to travel in one direction for ¼ second and stop within the button, or start moving from the button and move in a consistent direction for ¼ second. These approximations for the motion of a fingertip in the moments before and after it touches a button seem to be universal: almost every button touch either starts or ends that way. Interestingly though, many button touches only have one or the other of these motions. Of course, adjustments are needed in the case of people with tremors or other motor disabilities.

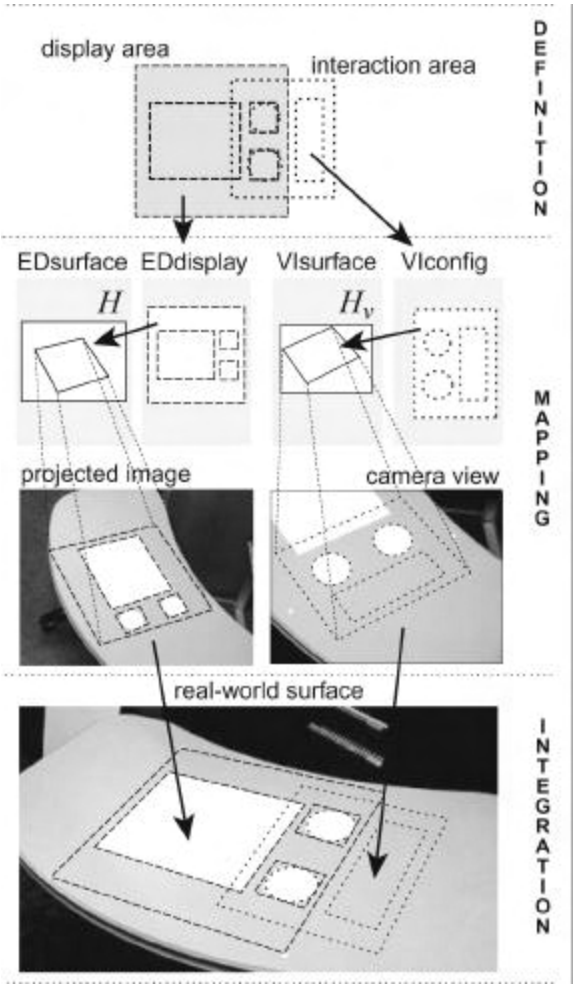

Figure 7. Process of creating an interactive surface on a table.

One of the interesting problems in this domain occurs when users occlude buttons as shown in Figure 5.b. Our experiments with users, even novices, have shown that they naturally avoid obstructing the projection if they are attending to it, but they are quite bad at staying out of the way when they are working with the physical world instead. Buttons have inherent resistance to triggering in these types of situations because any motion must pass several tests before it generates an event. It must resemble a fingertip for several frames; during that time it has to travel with a consistent direction and velocity and stop within a button. Such a sequence sometimes occur as a result of a chance combination of errors, contributing to a somewhat higher false positive rate for some buttons, particularly those which the user occludesfor a period of time rather than just passes through.

By using both the shape of the moving region as well as an analysis of its trajectory, this touch detection method gives far better accuracy than using either one alone. Because it relies directly on image data, rather than building an explicit model of the hand, it is computationally efficient enough to run at high frame rates on standard hardware.

Widgets can be combined in various ways in order to perform a complete interaction with the user. However, configurations of widgets are defined independently of the surface where they are used. That is, if an application interface using a set of buttons is moved to different surfaces, it is not necessary to define each button for each surface. Instead, the location of a widget within the camera image is determined at run time by appropriately mapping the configuration onto the camera image plane. In this case the surface refers to a set of camera parameters. These parameters, like those of the projector, have to be calibrated in advance. In particular, the calibration process has to determine the transformation H_v between the widget coordinate system and the camera image (as shown in Figure 4). Calibration of surfaces is performed using methods analogous to the 2nd and 3rd methods of projection calibration, although information about the expected location of the user, and the size of their hand, has to be obtained during an extra, special calibration step (see [31]for details).

3.3 Integrating Displaying and Sensing

The two preceding sections described the techniques and subsystems that provide the display (output) and sensing (input) functionality of the ED-projector. This section discusses the integration of these separate subsystems in a manner that affords application development and human interaction. One of our primary objectives here is to enable a specific user interaction to occur on any calibrated surface in a room without requiring the customization of this interaction for every surface. This leads to a clear separation of the abstract definition of an interaction from the actual surface upon which it occurs.

Our widgets are constructed using the standard Model View Controller (MVC) design pattern structure [33]. In our case, the projector subsystem is responsible for rendering the *view*, while the vision subsystem provides the *controller* input events. The *definition* layer of Figure 7 shows the separation of the display and interaction areas that are reponsible for the definition of the view and the interaction respectively. Display and interaction will often overlap as shown by the square overlapping buttons on the right of the *display area*. Provision is also made, however, for allowing interaction to occur in a region where nothing is displayed, as shown in the large rectangular region on the right of the *interaction area*. Notice that the display area here corresponds to the mesh M of the application plane decribed before (see also Figure 4).

Every interaction surface in a physical space is named and initially calibrated by the projection and vision subsystems yielding the transforms H and H_v respectively. H corrects the projected image while H_v provides the transformation to the camera image, for a specific surface as described in the prior sections. Figure 7 demonstrates the flow of definitions for display images and interaction data (widget data) from the *definition* layer to the *mapping* layer. During run-time, the system software automatically maps (through the warping functions H and H_v) the *definition* layer data into the projected image and the camera view associated with the current surface. The result is a projected image that generates application events when user gestures occur in the areas defined in the camera view, as shown in the *integration* layer of Figure 7.

4 PAINTING WITH INTERACTIVE GRAPHICS

To better understand how to use projected interactive graphics and how users relate to them, we have designed and implemented three application prototypes using the ED-projector. In the first application, an office space is augmented with dynamic wallpapers, visual notifications of messages, and computer access from multiple surfaces. The second prototype explores an assembly task where the instructions are projected directly onto the surfaces of the objects involved in the task. The third prototype is the "catch the frog" game described in the introduction to this paper.

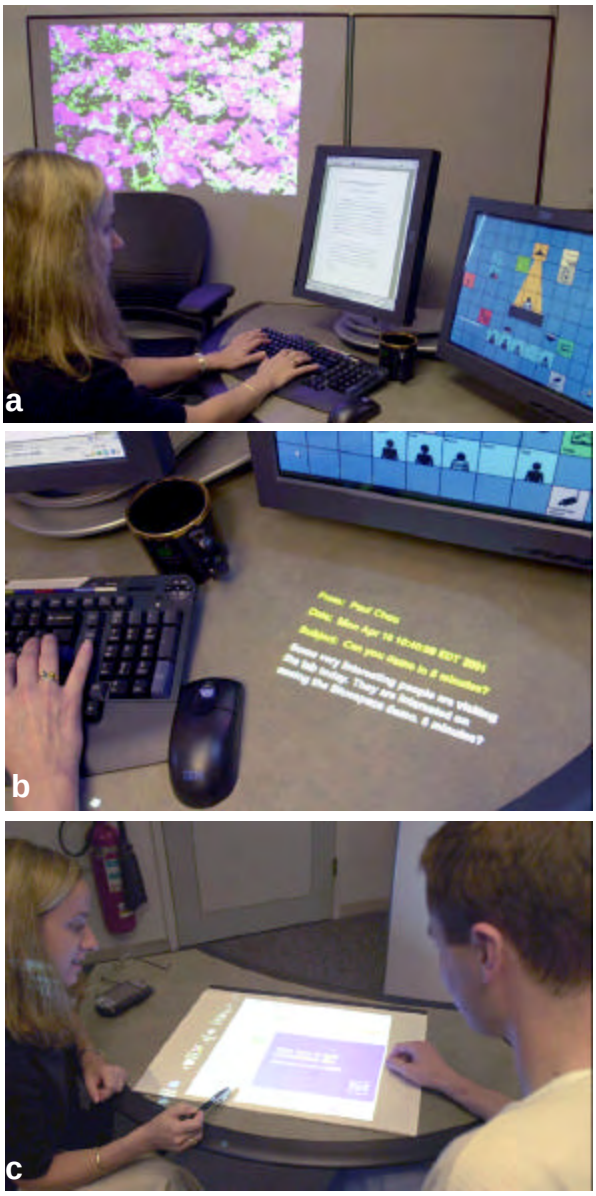

Figure 8. Uses of the ED-projector in an office application: a) to create dynamic wallpapers; b) to provide silent notification of e-mail; and c) to create a computer desktop on a working surface.

4.1 An Office Application

The first prototype application in which we have deployed the ED-projector is a futuristic personalized and context-aware workspace. The workspace has an 8' by 10' footprint and incorporates several sensors for measuring ambient lighting, temperature, humidity and noise level. Users and visitors to the workspace wear active badges that facilitate presence detection and identification. The desk chair is equipped with a pressure sensor connected to a wireless micro-controller, which detects if a person is sitting in the chair. This environment incorporates an ED-projector, strategically located to maximize coverage of the workspace. In this setting, we have used a non-interactive version of the ED-projector to facilitate greater personalization of the environment and to free its users from the confines of the desktop displays.

The presence of an ED-projector in the office tranfigures the space into an extended graphical display and enables the user to have a favorite painting on a wall, receive important notifications on surfaces that catch their attention, or use a tabletop as a display when collaborating with a colleague (see also the video figure). Figure 8.a shows an example of a dynamic wallpaper created by the ED-projector. The user can similarly have a virtual window that is connected to a live webcam looking at her favorite scenery (effectively making it a window office) or monitoring her child at home. The ED-projector can automatically switch to a neutral, less personal image when a co-worker comes to the office, based on the active badge.

Figure 8.b illustrates an urgent notification to the user appearing on a surface close to the user's current location. Here, the smart office detects that the user is seated at the desk and steers the notification right to the surface of the desk. By determining the position of the user, the system is assured that the user will see the urgent message without having to rely on the traditional sound alerts which tend to be disruptive to people working in a shared environment.

Another example of the utility of a steerable display is seen in Figure 8.c where two colleagues have created a computer desktop display on the working surface of their choice. Here, interactive capabilities, provided by a camera system, are planned to be incorporated to facilitate collaborative work.

4.2 Augmenting an Assembly Task

The ED-projector can be seen as a device for augmenting reality – a beam of light that can add information on to real world objects by highlighting them, adding text or images, flashing etc. For example, the ED-projector can show the user an object they are searching for, or indicate the procedure for an assembly task, and even where to place a part in an assembly.

We developed a prototype assembly task to highlight these themes of augmented reality and interaction anywhere. In this prototype application, which was experienced by hundreds of novice users in a major technical exhibition, the object to be assembled is a picture made of M&M's (multi-colored sugar-coated chocolates) where each M&M is regarded as a pixel of the picture.

The theme of interactivity anywhere is demonstrated several times in this prototype application (see also the video figure). Figure 9.a shows a white fabric on a table transformed into an interactive menu for color selection. The user simply touches the color of his choice to select it. Figure 9.b shows a "clickable" button appearing on a paint bucket that contains the M&M's of the selected color. Figure 9.b and c highlight the augmented reality and user assistance aspects of the system. In Figure 9.b the system highlights the bucket that contains the M&M's of the selected color, and also provides additional information on how many M&M's to pick. In Figure 9.c, the system points the exact places on the picture board where the M&M's should be placed. Figure 9.d shows another combination of interaction and augmented reality: the user moves her hand over the surface to interactively reveal the rest of the picture being built.

More than 650 people participated in this augmented assembly over 6 days and completed 4 different M&M pictures. Considering that the vision system and the projection system were integrated for the first time only in the weeks preceding the exhibition, the combined system worked remarkably well. A sample with 130 consecutive users with 621 button touch events (touching gestures or false detections) yielded correct detection of touching gestures in 81% of the events, with 12% false negatives and 7% false positives. If the bucket events are excluded from the count, the

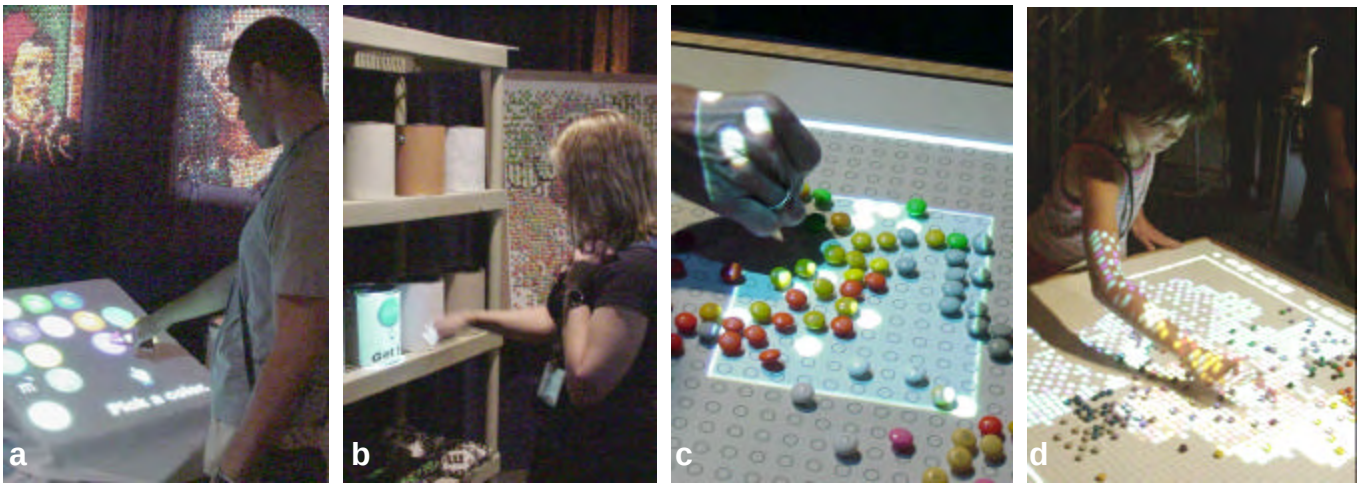

Figure 9. The M&M picture assembly task: a) choosing a color; b) "clicking" a bucket after getting the M&Ms; c) placing the M&Ms on the highlighted areas; and d) "finger painting" to reveal the complete picture.

performance exceeds 89%. A detailed account of the reasons why buckets yielded more errors is given in [31].

The experiment also revealed that users are readily able to relate to the notion of steerable interactive graphics. We also observed the value of the projection creating a shared experience between the users and the people observing the demo: in many cases, users learned the interaction process by first watching other users; and frequently, observers helped users suggesting what to do next. As mentioned before, a major advantage of projection over goggles is precisely this "social" nature of projection.

An interesting observation is that our users were less hesitant to interact with the fabric-covered interactive color selection menu (Figure 9.a) and the M&M placement board (Figure 9.c,d) than with the paint buckets (Figure 9.b). When they went to a bucket to pick up M&M's, many of them did not realize that they had to click the big "done" button on it (see also Figure 6.a). We had expected that, since this was the third occasion for touch interaction in the demo, users would quite naturally press the button on the bucket as instructed. Most of the time, this was not the case. To help the users in that situation we often gave them verbal instructions such as "Click the bucket," aiming to make clearer that there was, in fact, a button on the bucket.

4.3 "Fro…og!"

The ED-projector's unique characteristic of being able to quickly move the projected image to different surfaces is being explored in a computer game for young children. In this game, called "Fro…og!", an image of a frog is projected on a surface. Whenever the child tries to touch the frog, the camera detects her approach, and the frog "jumps" to another surface, that is, the projector moves the image to a wall or the top a chair.

Figure 1 shows two pictures of a user playing with the current prototype of "Fro…og!". The basic idea is to define surfaces in the environment where the animated frog can jump to. Whenever the user's hand approaches the frog, the next surface is randomly chosen, and an image of the frog jumping in that direction is briefly shown. After landing on the new surface, a simple animation of the frog jumping up and down is projected, so it becomes easier to find the frog in the environment.

In the next development phase of this game we plan to integrate it to a system that tracks the position of multiple children in the room, so the frog can jump to surfaces which are far from most players. Tracking is also particularly important to avoid projecting the frog directly onto the players which, in this scenario, can easily break the illusion of the game.

5 DISCUSSION

Having shown how we implemented a working prototype of an ED-projector and deployed it in three different applications, it is interesting to explore other possible applications and scenarios, and discuss how to design efficient interfaces for them.

5.1 Other Applications

We see two classes of applications of the ED-projector to create ubiquitous interactive graphics. The first involves the creation of computer interfaces anywhere, while the second deals with augmented reality in a variety of public and private spaces.

In the first class of applications, we regard the ED-projector as a generic input/output device that can replace, in many situations, current displays and interactive devices by creating computer desktop-like interactive displays on non-tethered surfaces. For example, a desktop application can be projected directly on any surface, as previously suggested by Wellner [9] and as demonstrated in subsection 4.1 . Unlike the interactive whiteboard described in [34], the projected application can be easily moved around the room, for instance, from a whiteboard on the wall to the top of a desk for more detailed reading. Similarly, the ED-projector can be used to bring information to the physical location where it is used or needed. For example, a database application managing reports can be projected on top of the file cabinet with hard copies of the reports.

In the second class of applications the ED-projector can augment reality by pointing to physical objects, showing connections among them, and projecting patterns to indicate movement or change in the real world. The prototype assembly task described in subsection 4.2 is a typical example of this class of applications. Another example applies to a library in which directions are provided to a user looking for a book by projecting arrows on the floor, walls and shelves, and finally by highlighting the desired book.

Steerable graphics produced by the ED-projector can also aid in collaborative work by providing large displays on convenient surfaces, and allowing easy reconfiguration of meeting spaces for different functions and teamwork styles, in similar ways to the office project described in subsection 4.1 .

The ED-projector can also be used to provide computer and information access in environments where traditional displays are

difficult to install, secure, or maybe unsafe to operate, such as public spaces and areas subject to harsh environmental conditions. The device also permits an interactive display to be brought to the proximity of a user, eliminating the need for the user to approach a traditional, fixed display. In particular, the ED-projector can facilitate the access and use of computers by people with locomotive disabilities. For instance, it can project an interactive display on a hospital bed sheet without requiring the patient to contact any device.

We also see the ED-projector as a potential enabler of a new generation of games where interaction happens not in the virtual world but in the physical world such as the "Fro…og!" game described above. Unlike games based on phones and portable computers [4], the use of the ED-projector provides high-resolution displays where characters and fantastic objects can move from surface to surface, creating a game that surrounds the players. Using video input, various user actions can be detected depending on the game needs, for example, hand gestures, body movements, foot action, and/or facial expressions. Above all, a single ED-projector installed on the ceiling of a living room or an arcade, using just software commands, can completely reconfigure the space to create different games according to the interests, age, and motor skills of the players.

The scenarios examined so far are chiefly concerned in creating ways for humans to command computers and/or obtain information. However, we see the projecting of interactive graphics ubiquitously as a way to provide computer agents the ability to "reach and touch" the real world and the people that inhabit it. By using projected patterns, it is possible to make computers not only interact with people, but act and coordinate them. For instance, a computer agent can control people in a line by projecting red lines and green arrows around the people on the line. A steerable projector system is, thus, able to create a "body", composed of light, for an agent. From this perspective, we may be creating an interface to the real world for computers.

5.2 Human Interface Issues

As foreseen by Weiser [1], ubiquitous interactive graphics do not seem to follow the standard paradigm of interaction based on the desktop metaphor. In our M&M demonstration, we had the opportunity to observe more than 650 novice users experiencing a scenario of true ubiquitous computing, in the context of an assembly task. On one side, we observed that they quite naturally accepted the idea of augmenting reality with information directly projected on objects. Moreover, we saw they naturally change their body's position to minimize problems like the shadow of their arms and body over the board. People seemed not only to accept projected graphics naturally but also to compensate for its limitations.

However, when trying to click a button, the users almost always applied pressure onto the projected surface with their fingers. Also, whenever a button touch was not detected by the vision system, the most common reaction was to press again with more pressure and, if still not successful, using the whole hand instead of a finger. It seems that people expect buttons, even virtual buttons, to react to pressure and not to touch, and to fail if enough pressure is not applied on them.

This observation clearly illustrates the limitations of the hand-as-a-mouse paradigm used in the M&M demo. Although it is easier for users to relate the ubiquitous experience to their everyday contact with desktop interfaces, the metaphor proved to be misleading to the true nature of the interface. Unlike a mouse that has appropriate haptic feedback for clicking gestures, touching a projected button is devoid of feedback movement. Similar observations prompt us to believe that a new paradigm for interaction with projected screens must be developed, with new kinds of widgets, tuned for the natural capabilities and limitations of projection and gesture-based interaction, just as the point and click paradigm is tuned for the characteristics of a physical mouse. Touch-screen-like interactions do translate better to the projected interface domain. Although our widgets are not based on surface contact, the widgets we have found useful so far (buttons, scroll bars and touch pads) are all based on motions people naturally make when touching.

One of the exciting aspects of this work is that it is an excellent environment to explore more adventurous gesture-based interaction styles. By giving the user feedback as to what gestural interactions the system is expecting and where in the environment they may be peformed, the cognitive load on the user can be reduced. Without such feedback a user would have to memorize what amounts to a set of complex dance steps to simply use a gesture-based interface.

This investigation of new human-computer interface paradigms is even more important in the case of multi-user interaction. Unlike desktops, wearable computers, or PDAs, projected interactive graphics naturally allow many users to simultaneously interact with an application. Although the applications we have developed so far are all single-user, it seems clear multi-user interaction will soon became an important area of research.

Another intriguing question is related to the different degrees of difficulty experienced by users when interacting with graphics projected on different supporting objects. Clicking a paint bucket seems to be harder to accept then clicking a table. Of course the design of our system and the simplicity of our experiment allow us neither to definitively state that the phenomenon does happen nor to draw conclusions about its causes. However, one hypothesis is that people seem to attribute different interactive capabilities to projected interactive graphics according to the nature of the surfaces themselves. Are the paint buckets "functionally fixed", that is, if something looks and acts like a paint bucket, is it less likely to be perceived as something else – namely, a touch-screen? We are currently preparing a study to determine the validity of this hypothesis.

Finally, our research so far has addressed only visual input and output. A promising area of research seems to be the application of the concept to steerability of other stimuli such as sound, using, for instance, the audio spotlights described by Pompei [35]. What kind of interfaces and applications can be created by a full multimodal steerable system?

6 CONCLUSION

This paper describes a new method and device for creating ubiquitously interactive graphics on non-tethered surfaces. We have shown three different methods to determine the warping function that corrects for oblique projection distortion and a computer vision system that tracks the user's hand over the projected images and detects touch-like gestures. We have also implemented three applications that realize ubiquitous computing concepts such as everywhere computer access and augmented reality. Based on the experience of running the M&M demo with hundreds of novice users, we have also detected a need to re-examine the current interaction paradigms, and possibly, investigate and propose new ones.

We see this work as an important contribution to realize the vision of ubiquitous computing, without asking people to wear headsets or wiring the surfaces of everyday objects. However, we do not advocate projection-based solutions as a panacea for ubiqutious applications: in many cases, it is more desirable and efficient to use head-mounted displays. A good analogy is to

compare headsets and projectors to Walkmans and stereo sound systems. Walkmans are unsurpassable as a way to provide music when people are moving around or in public spaces such as trains and airplanes. They are very effective in creating a personal experience, albeit not able to create social interaction. Similarly, we keep loving stereo systems because they render music to a whole space, nurturing a collective experience that is attached to the space itself. We see similar advantages and limitations when comparing headsets and steerable projectors as a way to render interactive graphics onto the real world.

Acknowledgements

"M&M" is a registered trademark of Mars, Inc. "Walkman" is a registered trademark of Sony Corporation.

References

1. Weiser, M., *The Computer for the Twenty-First Century.* Scientific American, 1991. **265**(3): p. 94-100.
2. Pinhanez, C. *The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces*. in *Proc. of Ubiquitous Computing 2001 (Ubicomp'01)*. 2001. Atlanta, Georgia.
3. Barfield, W. and T. Caudell, eds. *Fundamentals of Wearable Computers and Augmented Reality.* 2001, Lawrence Erlbaum Assoc. 728.
4. Raskar, R., et al. *The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays*. in *Proc. of SIGGRAPH'98*. 1998. Orlando, Florida.
5. Ishii, H. and B. Ullmer. *Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms.* in *Proc. of CHI'97*. 1997. Atlanta, Georgia.
6. Bellour, R., et al., *Tony Oursler*. 2001: Ediciones Poligrafa. 208.
7. Naimark, M., *Spatial Correspondence in Motion Picture Display.* SPIE Optics in Entertainment II, 1984. **462**: p. 78-81.
8. Bolt, R., *Put That There: Voice and Gesture at the Graphics Interface.* ACM Computer Graphics, 1980. **14**(3): p. 262-270.
9. Wellner, P., *Interacting with Paper on the DigitalDesk.* Communications of the ACM, 1993. **36**(7).
10. Morishima, S., et al. *HyperMask: Talking Head Projected Onto Real Objects*. in *Proc. of Multimedia Modeling (MMM'00)*. 2000: World Scientific.
11. Raskar, R., et al. *Shader Lamps: Animating Real Objects with Image-Based Illumination.* in *Proc. of 12th Eurographics Workshop on Rendering*. 2001. London, England.
12. Krueger, M.W., *Artificial Reality II*. 1990: Addison-Wesley.
13. Keays, B. and R. Macneil. *metaField Maze.* in *Proc. of SIGGRAPH'99*. 1999. Los Angeles, California.
14. Omojola, O., et al., *An Installation of Interactive Furniture.* IBM Systems Journal, 2000. **39**(3&4): p. 861-879.
15. Ju, W. *Origami Desk*. in *Proc. of SIGGRAPH'01 - Conference Abstracts and Applications*. 2001. Los Angeles, California.
16. Back, M., et al., *Designing Interactive Reading Experiences for a Museum Exhibition.* IEEE Computer Magazine, 2001. **34**(1): p. 1-8.
17. Underkoffler, J., B. Ullmer, and H. Ishii. *Emancipated Pixels: Real-World Graphics in the Luminous Room*. in *Proc. of SIGGRAPH'99*. 1999. Los Angeles, CA.
18. Rekimoto, J. *A Multiple Device Approach for Supporting Whiteboard-based Interactions*. in *Proc. of CHI'98*. 1998. Los Angeles, CA.
19. Schiele, B. and J.L. Crowley, *Recognition without Correspondence using Multidimensional Receptive Field Histograms.* IJCV, 2000. **36**(1): p. 31-52.
20. Wu, Y. and T. Huang, *Vision-Based Gesture Recognition: A Review.* Lecture Notes in Artificial Intelligence, 1999. **1739**.
21. Hardenberg, C.v. and F. Berard. *Bare-hand human-computer interaction*. in *Proc. of Workshop on Perceptive User Interfaces, PUI'01*. 2001. Orlando, Florida.
22. Quek, F., T. Mysliwiec, and M. Zhao. *Finger mouse: A freehand pointing interface*. in *Proc. of International Workshop on Automatic Face- and Gesture-Recognition*. 1995. Zurich, Switzerland.
23. Segen, J. *GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction.* in *Proc. of ACM Multimedia Conference*. 1998. Briston, England.
24. Kjeldsen, F., *Visual Recognition of Hand Gesture as a Practical Interface Modality*. 1997, Columbia University: New York, New York.
25. Raskar, R., et al. *Multi-Projector Displays Using Camera-Based Registration.* in *Proc. of IEEE Visualization'99*. 1999. San Francisco, CA.
26. Pinhanez, C., F. Nielsen, and K. Binsted. *Projecting Computer Graphics on Moving Surfaces: A Simple Calibration and Tracking Method.* in *Proc. of SIGGRAPH'99*. 1999. Los Angeles, California.
27. Raskar, R. *Oblique Projector Rendering on Planar Surfaces for a Tracked User*. in *Proc. of SIGGRAPH'99*. 1999. Los Angeles, California.
28. Faugeras, O., *Three-Dimensional Computer Vision: A Geometric Viewpoint.* 1993, Cambridge, Massachusetts: The MIT Press.
29. Sukthankar, R., R. Stockton, and M. Mullin. *Smarter Presentations: Exploiting Homography in Camera-Projector Systems*. in *Proc. of ICCV'01*. 2001. Vancouver, Canada.
30. Yang, R. and G. Welch. *Automatic and Continuous Projector Display Surface Calibration Using Every-Day Imagery.* in *Proc. of 9th International Conf. in Central Europe in Computer Graphics, Visualization, and Computer Vision*. 2001. Plzen, Czech Republic.
31. Kjeldsen, R., et al. *Interacting with Steerable Projected Displays*. in *Submitted to Face and Gesture'02*. 2002.
32. Wren, C., et al., *Pfinder: Real-Time Tracking of the Human Body.* IEEE Trans. Pattern Analysis and Machine Intelligence, 1997. **19**(7): p. 780-785.
33. Buschmann, F., et al., *A System of Patterns - Pattern Oriented Software Architecture.* 1996, New York, New York: John Wiley & Sons.
34. Crowley, J.L., J. Coutaz, and F. Berard, *Things that See.* Communications of the ACM, 2000. **43**(3): p. 54-64.
35. Pompei, F.J. *The Use of Airborne Ultrasonics for Generating Audible Sound Beams.* in *Proc. of 105th Audio Engineering Society Convention*. 1998. San Francisco, CA.