Moveable Interactive Projected Displays Using Projector Based Tracking

Johnny C. Lee[1,2], *Scott E. Hudson*[1], *Jay W. Summet*[3], *Paul H. Dietz*[2]

[1]Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
{*johnny*, *scott.hudson*} @cs.cmu.edu

[2]Mitsubishi Electric Research Labs
201 Broadway, Cambridge MA, 02139
dietz@merl.com

[3]GVU Center & College of Computing
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332
summetj@cc.gatech.edu

ABSTRACT

Video projectors have typically been used to display images on surfaces whose geometric relationship to the projector remains constant, such as walls or pre-calibrated surfaces. In this paper, we present a technique for projecting content onto moveable surfaces that adapts to the motion and location of the surface to simulate an active display. This is accomplished using a projector based location tracking techinque. We use light sensors embedded into the moveable surface and project low-perceptibility Gray-coded patterns to first discover the sensor locations, and then incrementally track them at interactive rates. We describe how to reduce the perceptibility of tracking patterns, achieve interactive tracking rates, use motion modeling to improve tracking performance, and respond to sensor occlusions. A group of tracked sensors can define quadrangles for simulating moveable displays while single sensors can be used as control inputs. By unifying the tracking and display technology into a single mechanism, we can substantially reduce the cost and complexity of implementing applications that combine motion tracking and projected imagery.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. H5.1 [Multimedia Information Systems]: Augmented Reality.

General terms: Measurement, Human Factors

Keywords: projector based tracking, augmented reality, simulated displays, physical interaction

INTRODUCTION

Since the mid-1990's, computer driven data projection has become a staple display technology in both commercial markets and technology research projects around the world. This has lead to the availability of lower-cost and better performing projection systems and a vast library of exploratory applications that reach well beyond simply showing presentations or movies on a wall. Research such as [13, 15, 16] provide visions of future work offices that use projectors to transform large surfaces in our environment to dynamically suit the needs of our task. [9, 11, 14] describe specific techniques for using projection to augment the appearance of objects for aesthetic customization or for information display. These techniques bring popular visions of future ubiquitous computing environments closer to becoming a practical reality. However, they typically require that the geometric relationship between the projector and the display surface remain static after an initial calibration. As a result, these advanced surface augmentation techniques have been primarily reserved for stationary objects such as walls, furniture surfaces, and floors.

[1, 12, 20] have explored using tracking technologies to project content that is motion-matched to a moveable surface augmenting its appearance. However, this work has primarily depended on either electromagnetic or vision based tracking systems that add a substantial amount of infrastructure, cost, and complexity to achieve the desired affect. Additionally, these approaches still require an often complex pre-calibration step to map the tracking data to the projected image.

In this paper, we present a technique for projecting motion-matched content onto moveable surfaces without the need for an external tracking technology. This is accomplished by using the projector to provide display functionality as well as location tracking simultaneously. By unifying the tracking and display technology into a single mechanism, we can eliminate the cost and infrastructure necessary for external tracking systems, as well as greatly simplify the implementation and execution of applications that combine motion tracking and projected imagery. In Figure 1, we show an example application that uses a ceiling mounted projector and an ultra-lightweight, low-cost, instrumented surface to simulate a functional tablet display. Four optical sensors and a resistive touch screen surface are mounted onto a foam-core board. The surface is only slightly heavier than a standard clipboard and a small fraction of the cost of a tablet PC, yet allows the user to move and interact with the surface in a tablet PC-like manner.

To achieve this effect, the simulated tablet surface contains optical sensors at each corner. When positioned within the projection area, we can project a sequence of patterns to recursively divide the projection area allowing each sensor to discover its location in the projector's screen space. This location data is then reported back to a host computer that

UIST '05, October 23–27, 2005, Seattle, Washington, USA.

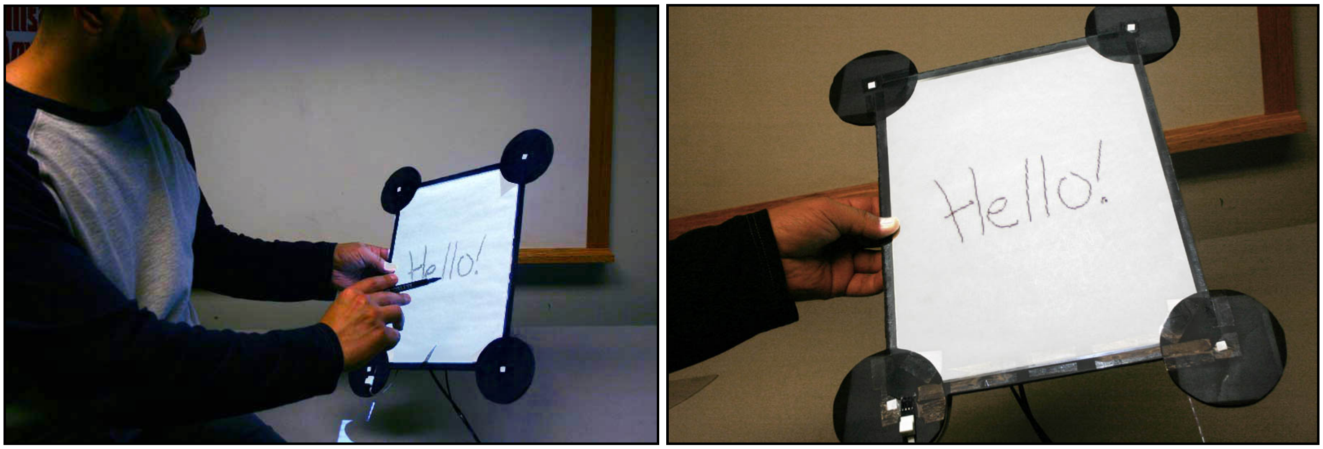

Figure 1 - A handheld surface containing four optical sensors and a touch sensitive surface is tracked in real-time using projector based tracking. The location data is then used to warp the projected content to match the surface movements simulating an ultra-lightweight tablet display at very low-cost.

is able to warp the projected content to fit the boundaries of the target surface as defined by the location data from the sensors. This result in the ability to create simulated displays on a tablet as shown in Figure 1, as well as a number of other applications and interaction techniques that combine motion-tracking and projected imagery described later in this paper.

RELATED WORK

As mentioned above, projecting motion-matched content onto moveable surfaces has been explored by a number of previous research projects. In particular, the Dynamic Shader Lamps work [1] was one of the first to explore augmenting the appearance of surfaces with projected light whose geometric relationship to the projector did not remain fixed. Users could pick-up and manipulate the decorated object as well as use a tracked stylus to color the surface with virtual paint. The geometric relationship between the projector and the target surface was acquired using a manual pre-calibration step and is then maintained throughout the interaction with a six-degree of freedom electromagnetic tracking system. In [12], we revisited this type of geometric freedom, but in the form of allowing a user to manipulate a hand-held projector rather than moving the projection surface. This was accomplished using visually distinct markers and a projector-mounted camera. [20] used active LED makers and a high-quality video camera for tracking. Though [22] does not specifically mention use with a projector, it describes an elegant camera-based technique for tracking and interacting with quadrangles without the need for visually distinct markers. Quadrangle tracking is useful for supporting interactions that are compatible with existing 2-D graphical user interface systems or simulating moveable rectangular displays. However, these approaches require an external tracking technology that must be calibrated to match the projected content. These tracking technologies are typically costly (hundreds to thousands of US dollars) and require a cumbersome support infrastructure. Additionally, the initial calibration process can be tedious and complex.

In [10], we presented a technique for discovering the locations of optical sensors in the projection area by using a sequence of black and white Gray-coded binary patterns. This technique provides the basic foundation of using a projector as a location discovery technology as well as a display device. However, our previous work did not support simultaneous tracking and content projection. In [17], we introduced using separate global discovery patterns and localized relative-offset tracking patterns to allow simultaneous image projection and motion tracking with a single projector. This work explored alternative patterns that provided variable resolution tracking of a single sensor and error checking. The tracking speed and accuracy varied depending on the magnitude of the sensor velocity. In this work, we describe a hybrid approach that provides pixel-accurate location tracking with a constant interleaved *x*-*y* update rate of 12Hz as well as discuss several issues that arise when tracking multiple sensors simultaneously.

Our previous work used high-contrast black and white patterns for location discovery. However, these patterns are easily visible by human observers resulting in visual stress and distraction potentially detrimental to the user experience in an interactive application. To address this problem, we have developed a method for projecting tracking patterns in a low-perceptibility manner using a modified commercial projector combined. Embedding low-visibility patterns in a projected image was pervious explored in [5]. However, that approach relied on a costly synchronized high-speed camera system for pattern detection. Our approach uses a sensing package costing less than $10USD.

REDUCING PATTERN PERCEPTIBILITY

In our previous work, the discovery and tracking patterns consisted of full white and full black regions that utilized 100% of the projector's dynamic range to improve signal detection at the receiving end. Though sufficient for accomplishing the primary task of discovering the sensor locations, these high-contrast patterns may result in a substantial amount of visual stress and discomfort to nearby viewers when projected at 60Hz, which is the typical refresh rate of commercially available video projectors.

Ideally, we would like to project these patterns at high-speeds in infrared making the location discovery process entirely imperceptible to a human observer. The imaging technology used inside commercially available Digital Light Processing (DLP) projectors is capable of achieving this goal while simultaneously preserving the traditional display capabilities of the projector [21]. However, to harness such capability requires building a fully custom Digital Micro-Mirror Device (DMD) based projector from raw components. This would involve substantial amounts of development resources and proprietary expertise. Though we are exploring this path, we have developed an intermediate solution to achieve low-perceptibility tracking patterns using a modified commercial DLP projector. This allows us to explore basic interaction techniques with an early "proof of concept" device without necessarily requiring a high-speed infrared projection device.

The pure white and black patterns used in [10, 17] delivered a data stream to each sensor in a manner analogous to an amplitude shift keyed (ASK) transmission. The amplitude modulation corresponds to the difference in white and black intensities and the effective carrier frequency is the cumulative frequencies found in the bandwidth of light emitted by the projector. With such a wideband carrier in the visible light spectrum and a modulation rate of 60Hz, the patterns are easily visible by a human observer. Our intermediate solution to this problem is to transmit the data using frequency shift keyed (FSK) modulation rather than ASK modulation. We take advantage of the imaging process used in commercial DLP projectors to achieve a FSK transmission alternating between carrier frequencies of 180Hz and 360Hz. The result is a tracking pattern that appears to be solid gray to a human observer but in fact contains rapidly changing location data detectable by a light sensor. The data modulation rate still remains 60Hz, but our human vision system is not able to detect a difference between the two carrier frequencies thus making the modulation imperceptible. To explain how this is accomplished, we must first briefly describe how consumer grade single-chip DLP projectors work.

The core imaging components of a DLP projector include a high-intensity lamp, a rotating color wheel, a DMD chip, and a projection lens. Light from the lamp is directed toward the surface of the DMD, which modulates the direction of reflected light using a high-density array of rapidly actuated mirrors. Each mirror on the DMD corresponds to a single pixel in the projected image whose intensity is determined by the positional duty cycle of the actuated mirror. To create a color image, a rotating color wheel is placed in front of the lamp to quickly cycle between red, green, and blue light sources. The DMD processes each separate color channel of the source image sequentially. Typically, the color wheel spins at either two or three times the base refresh rate of the video signal of 60Hz. The human vision system integrates the images back together to create the appearance of a single color image [21]. Some details of this imaging process have been intentionally omitted since they are not critical to understanding the basic technique.

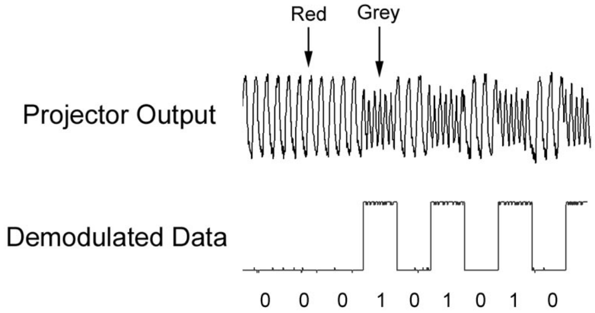

Figure 2 – Frequency shift keyed data transmission using two colors and a modified DLP projector.

To achieve the FSK modulation described above, we removed the color wheel from an InFocus X1 DLP projector, which contains an 800x600 pixel (SVGA) resolution DMD. This creates a gray-scale only projector, which is now capable of generating multiple versions of gray that have the same apparent intensity to the human eye, but have very different signal patterns when viewed by a low-cost optical sensor. The color space of the projector is flattened allowing us to generate the same apparent intensity in multiple ways. Specifically, we select either a bright red or a medium gray that, when rendered by our modified projector appear to be a similar intensity of gray to a human observer, but are manifested as a 180Hz signal and a 360Hz signal respectively, Figure 2. A noteworthy implementation specific detail is that when we removed the color wheel from the InFocus X1, the internal color calibration system resulted in an unexpected loss of the blue channel. This is why the gray color is manifested as only a 360Hz signal rather than a 540Hz signal. Understanding the exact cause of this side effect requires a detailed understanding of the design of the InFocus X1. But because it did not fundamentally affect the approach, it was not deeply investigated.

By using these two colors, we can hide the tracking patterns in what appears to be mostly solid gray squares. In our implementation, the gray regions retain a slight perceptible flicker. This is an artifact introduced by the projector's internal color processing system managing the transition between the two colors resulting in a minor deviation from the carrier frequencies between frames. As a result, the transitions appear momentarily brighter or darker than either base color. However, the flicker is very subtle and is not likely to be a substantial visual distraction when performing a task.

ACHIEVING INTERACTIVE RATES

As described in [10], the number of Gray-coded binary patterns necessary to resolve the location of a light sensor to a single pixel in a projection area is equal to $\log_2$(number of pixels). Thus, an SVGA (800x600) projector requires 20 images. Using 60Hz modulation, this yields a maximum update rate of only 3Hz assuming no overhead for sensor synchronization patterns. Since the amount of sync overhead is typically implementation specific, we will assume it is negligible for the purposes of this description and then revisit this issue in more detail later in “Implementation Details”.

We can improve this update rate by using a separate full-screen location discovery step followed by localized tracking with smaller patterns. Once we discover the absolute position of each sensor, we can project smaller tracking patterns over their locations to obtain incremental offsets. Smaller patterns require fewer divisions to resolve down to a single pixel. Therefore, we can acquire incremental offsets much faster than absolute positions. Additionally, small, localized tracking patterns liberate the rest of the projection area for display purposes.

In our current implementation, we use square tracking patterns centered over each sensor that subdivides the contained space horizontally five times and vertically five times using Gray-coded FSK binary patterns. This creates a 32x32 unit grid centered over the previous sampled location of the sensor. Once the offset is found, the tracking pattern is then re-centered over the updated location. The number of subdivisions for the localized tracking patterns was chosen primarily for its even division into 60Hz yielding an *x*-*y* coordinate pair update rate of 6Hz. Finer or coarser tracking patterns could be selected for speed and accuracy depending on the needs of the target application. However, there is a limitation on the minimum number of divisions a particular implementation can support due to system latency.

Latency and Interleaving

When using an incremental tracking approach, it is important that we use the most recent available location data of each sensor to properly position the tracking patterns for the next offset sample. The older the location data is, the less likely it is to be an accurate estimation of the current sensor position and the less likely it is for the sensor to still be in the tracking area. The result of the current sampling should be used to position the next immediate sampling. Thus, incremental tracking is actually a real-time positional feedback loop whose performance is primarily determined by sample rate and feedback latency. Our sample rate, as described above, is 6Hz for acquiring both the *x* and *y* offsets using 10 patterns. The feedback latency is the time delay between issuing the draw command in software and the execution of the reader thread that parses the corresponding data. This is a function of the scheduling algorithm in the host computer’s operating system, the video rendering subsystem, the projector’s video processing circuitry, the sensor demodulation algorithm, and the speed of light (negligible in most cases). In our implementation, we observed an average latency of ~60ms which corresponds to 3-4 frames at 60fps. Since we only use 10 frames per tracking sample, a latency of 4 frames is a substantial increase to the overall sensing time. A large portion of this latency is caused by task scheduling within the operating system of the host PC and is not inherent to our tracking technique.

To prevent this latency from severely impacting our tracking rate, we take advantage of the Gray-coded patterns which resolve the *x* and *y* offsets independently. This axis independence allows us to use an interleaved tracking technique. This effectively pipelines the tracking operations allowing us to transmit the tracking patterns for one axis while we wait for the result for the other axis to return to the tracking software. Since the feedback latency is less than 4 frames and the patterning time for a single axis is 5 frames, we can retain 100% utilization of the projector’s tracking capability. The end result is a tracking update rate of 12Hz alternating between each axis. It is important to note that though we were able to find a reasonable solution using grouped Gray-coded patterns, feedback latency places a substantial constraint on future exploration of alternative patterns that may utilize recent sensor data to improve tracking performance. Tracking algorithms that require instantaneous or near instantaneous feedback from sensors are not likely to be executable in practice.

Localized Pattern Size and Shape

The size and shape of the localized tracking patterns play a critical role in determining the range of movements supported by this tracking technique. If the sensors move outside of the tracking pattern boundaries within the sampling period, the sensor will become lost requiring a full-screen sensor re-discovery process. This requires a momentary interruption (0.367secs in our implementation) of an application’s projected content and thus should be avoided. The size, shape, and sample rate of the localized patterns determine the maximum sensor velocity the system can continuously track without error.

We have described our tracking patterns thus far as resolving to an offset within a 32x32 unit grid using five horizontal patterns and five vertical patterns. In the simplest implementation, this grid might be mapped to a 32x32 pixel area in the projected image. This may provide an acceptable coverage of movements for applications that primarily focus on tracking objects in the image plane or tracking single sensors. However, if the distance between the sensors and the projector is allowed to change substantially, a fixed pixel dimension of the patterns will result in a wide variation in the maximum supported tracking velocity in terms of meters per second. This can be problematic and confusing to the user, for example, when moving surfaces that are meant to be hand-held such as a simulated tablet, shown in Figure 1 and 6.

For these applications we use a fixed physical size for the tracking patterns to maintain a consistent maximum tracking velocity regardless of distance from the projector. This is accomplished by using the known geometry of the display surface and the currently observed locations of the corners. Using fixed physical dimensions also maintains the relative size of the tracking patterns with respect to the physical display as well as the projected content. Additionally, it produces a variable pixel accuracy behavior based on distance. As the display moves farther from the projector, the tracking patterns will shrink in pixel space resolving down to a single pixel. As the display moves closer to the projector, the pixel density increases making pixel-perfect alignment less important and the accuracies of the tracking patterns reduce accordingly.

The shape of the tracking patterns we use in our implementation are simple squares aligned to the image plane of the projector. We use this shape because of the axis-aligned nature of the Gray-code patterns. Elongated shapes could be used to permit a higher range of movement in one particular direction for applications such as a projected slider widget. Similarly, a variety of pattern geometries could be used to track specialized sensors that have restricted or expected ranges of movement for application specific tasks or interaction techniques. However for general purpose tracking in two-dimensions, a shape with a greater degree of radial symmetry, allowing a similar freedom of movement in any direction, is more appropriate.

Motion Modeling

It is possible to soften the maximum supported tracking velocity constraint by modeling the motion of the sensors to predict likely future locations. Since physical motions exhibit a high degree of temporal continuity, recent motion history can be used to generate a strong prediction of likely positions in the near future. The model we use consists of a moving average of recent velocity, acceleration, and jerk (derivative of acceleration). Combining these values and the most recent sampled position, we can calculate a probable path for the sensor and then center the tracking pattern accordingly. Fortunately, the predicted locations do not need to be exact since the tracking patterns search over an area giving the system a relatively large acceptable margin of error. By using a motion model, we can adjust the locations of the tracking patterns to dramatically increase the range of movements the system can successfully track. The motion constraint is then moved to the third derivative of position, jerk. The model can be made to include further derivatives or otherwise be made more complex. However, in our exploration this simple model provided a good balance between the coverage of the motions used in our test applications and tracking errors due to mis-prediction. Mispredictions are an inherit risk of any predictive model, since no model can accurately account for all the complexities of the physical world or the intentions of the user. Motion models can be selected and tweaked to adjust the balance between freedom of movement and tracking failures. The appropriate balance will be application and implementation specific.

Tracking Loss Strategies

Tracking loss can occur for several reasons including exceeding the supported motion constraints, model mis-predictions, and corrupt or unavailable tracking data. In some cases, circumstances may allow the system to reacquire the sensor from a momentary tracking loss through chance. However, if a sensor is identified as being conclusively lost, a fallback strategy is necessary to re-discover the sensor locations. This may be triggered manually through user input, or by a pre-defined timeout for lack of sensor data, or possibly signaled by a sequence of erratic improbable offsets (sometimes a symptom of interference). There are several options that can be employed for recovering lost sensors, each having their own advantages and disadvantages with no clear choice as to which is the best overall behavior for all applications. In this section, we describe recovery strategies when tracking only a single sensor. If multiple sensors with a known geometric relationship are tracked simultaneously, this information can be used to make informed predictions and will be discussed later in "Occlusion Detection and Behavior."

The simplest option is to perform a full screen discovery process to search the entire projection area for lost sensors. The downside is that the entire projection area becomes gray, interrupting any projected application content. However, the upper bound on the recovery time can be as short as 1/3rd of a second assuming the sensors remain in the projection area. If the conditions of use result in relatively infrequent sensor loss, this may be a reasonable strategy and is the one we use in our current implementation.

Another approach described in [17] is to grow the tracking patterns around the last known valid location until it contains the sensor again, shrinking back to normal size after the correct location has been discovered. This has the benefit of searching only a small region of the projection area yielding a potential recovery time shorter than 1/3rd of a second as well as causing a minimal amount of obstruction to any projected content. However, the upper bound on the recovery time is determined by the growth function and may result in an average performance substantially longer than the time needed to perform a full-screen discovery. Additionally, the expansion and contraction increase the visual saliency of the tracking patterns, which may potentially be more distracting and detrimental than a momentary gray screen. Alternatively, historical or statistical approaches can be employed to determine probable locations of a lost sensor. However, these techniques also suffer from high upper bounds on recovery time and increased visual saliency caused by frenetic pattern movement. Preferable behavior will likely depend on the application, usage environment, and the specifics of the implementation.

OCCLUSION DETECTION AND BEHAVIOR

In addition to reducing the perceptibility of the tracking patterns, FSK based transmission also improves our ability to detect sensor occlusion over our previous ASK based transmission. In an ASK transmission, it is often impossible to distinguish the difference between signal loss and a long sequence of '0' bits. When using FSK, the lack of either carrier signal signifies that the connection has been lost. Additionally, our FSK technique uses very narrow band carrier frequencies when compared to the white and black image ASK transmissions used in our prior work. This makes it easier to filter out interference and reject corrupted bits. These properties allow us to detect occlusions and other signal errors on a per-bit basis providing highly robust behavior. When using projector based tracking for interactive surfaces such as the one shown in Figure 1, sensor occlusions may occur frequently. Per-bit detection of signal loss allows an occlusion to occur at any point in the tracking period without resulting in a tracking failure due to corrupted data. Though reasonably robust detection of signal loss can be accomplished with ASK transmission using trailing check bits [17], this additional data reduces the overall update rate and does not guarantee detection.

To properly demodulate a FSK transmission typically re-

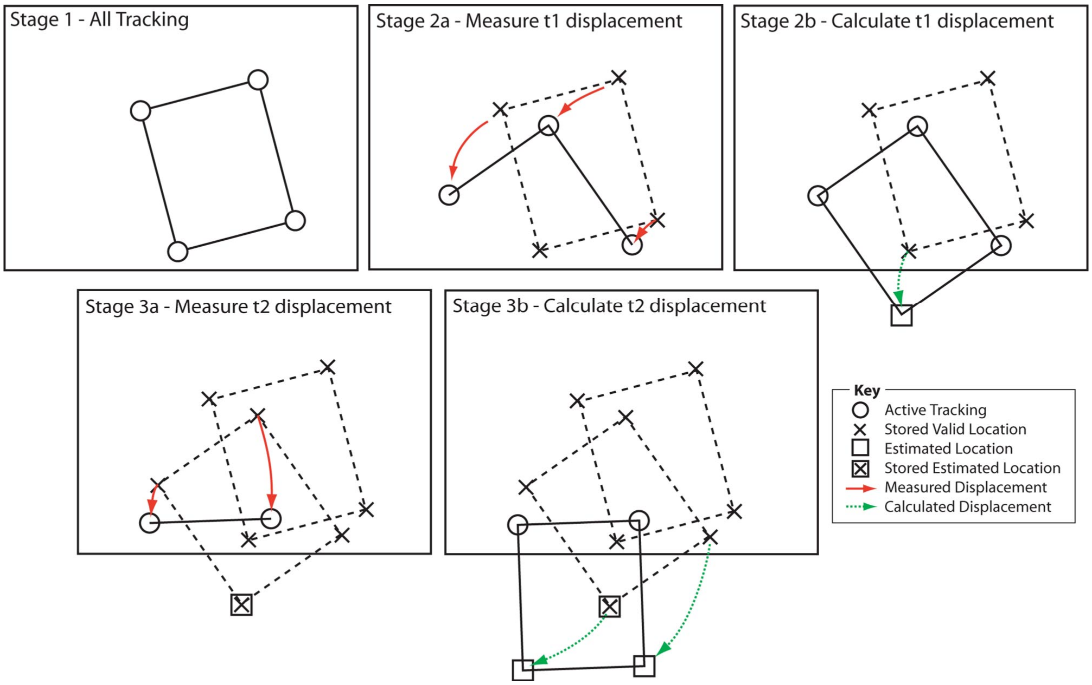
Stage 1 - All Tracking
Stage 2a - Measure t1 displacement
Stage 2b - Calculate t1 displacement
Stage 3a - Measure t2 displacement
Stage 3b - Calculate t2 displacement
Key
Active Tracking
Stored Valid Location
Estimated Location
Stored Estimated Location
Measured Displacement
Calculated Displacement

Figure 3 – Illustration of the sensor location estimation process when dealing with 0, 1, and 2 unavailable sensors.

quires either analog filtering electronics or sufficient computing power to perform real-time signal processing. However, these substantially increase the cost and complexity of the sensor design. In our implementation, we use a simple software demodulation scheme that tracks signal amplitude and edge counts. Though a crude approximation of proper FSK demodulation, it can be run on a low-cost microcontroller with minimal external components and has worked effectively in our explorations. A transmission error is defined as a sudden drop in signal amplitude, insufficient signal amplitude, or invalid edge count. These errors are able to flag signal loss due to occlusions or leaving the projection area and some limited forms of signal interference. The carrier frequencies of 180Hz and 360Hz generate 6 and 12 edges respectively every frame period. Valid edge counts (using a +/- 1 margin) are converted into 0's and 1's while invalid edge counts are flagged as errors. These error flags are transmitted back to the host computer with the decoded bit string.

Once we are able to reliably identify these transmission errors, we must decide what policy to use in the behavior of the tracking patterns when the sensor location is unavailable. One policy is to simply discard the data and reuse the last known valid position of the sensor. The resulting effect is that the tracking pattern does not move if an occlusion occurs. When tracking individual sensors, this may be the most appropriate policy. In our exploration, we informally observed that many occlusions occur when the user is attempting to interact with other objects rather than moving the sensor itself, such as pointing at the projected content, drawing on the touch sensitive surface, or just walking in front of the projector. Thus, the likelihood that a sensor remains stationary during an occlusion is reasonably high.

If we are tracking multiple sensors simultaneously in a known geometric configuration, such as the simulated tablet application shown in Figure 1, we can use the displacement of the available sensors to generate an estimated location of any occluded or off-screen sensors. With respect to the execution of this estimation technique, there is no functional difference between sensor occlusion and a sensor moving out of the projection area. Thus, for the purposes of explanation, we will describe this process in the context of a tablet exiting the projection area as illustrated by Figure 3. In Stage 1, all sensors are visible by the projector and no estimations are necessary. If one sensor moves outside of the projection area, Stage 2a, we store a snapshot of the last valid locations for all four sensors and then measure the displacement of the three tracked sensors. These six offset values can be used to calculate the top six values in a 3x3 affine transformation matrix, *t1*. The estimated location is then the last valid location of the lost sensor multiplied by *t1*, Stage 2b. This affine transform encapsulates translation, rotation, scale, and skew providing a very strong estimate of the lost sensor's location. Though tracking may be impossible if the estimated location is outside of the projection area, this estimated point can still be used to preserve the geometry of the projected content. When a second sensor is lost, Stage 3a, another snapshot is taken of all four sensor locations (tracked or estimated) at the time of disappearance. Then the displacement of the remaining two sensors from their respective snapshot locations is used to generate another transform *t2*. However this transform is significantly simpler than *t1* encapsulating only two dimensions of translation, one degree of rotation, and one degree of scale. As expected, the strength of the estimation becomes progressively weaker as we have fewer sensors to compute the transformation. If an additional sensor is lost and we are left with a single actively tracked sensor, we are limited to only updating the translation of the geometry to motion-match the remaining corner. The screen must be brought back into the projection area at a similar orientation if the tracking patterns are to re-acquire the lost sensors.

In our exploration, we found this occlusion behavior to be effective at estimating sensor locations under typical usage. However, performing complex movements when tracking data is scarce will cause the estimations to be incorrect. If this occurs, a full-screen discovery or another fallback strategy described in "Tracking Loss Strategies" must be performed.

A closer look at Stage 3b shows that final estimated location of the first lost sensor is actually the result of two transformations, *t1* and *t2,* from the last known valid location. This is significant because the estimation error of each transform is compounded defining a relationship between likelihood of estimation error and the order in which a sensor was lost. Additionally, you can see in Stage 3b that we specifically transform a stored snapshot of the estimated location rather than calculate the final estimated value dynamically using *t1* and *t2*. The reason for doing this is because we are not guaranteed to have LIFO ordering of sensor loss and re-acquisition. Otherwise, we could simply implement a matrix stack for each lost sensor and push and pop matrices as needed. However, when LIFO ordering is not maintained, matrices may have to be deleted or modified in the middle of the stack. Using location snapshots simplifies the implementation and accounting tasks required to support non-LIFO ordering of sensor loss and re-acquisition.

IMPLEMENTATION DETAILS

The projector used in our implementation is an InFocus X1 DLP projector, which provides an 800x600 (SVGA) image at a 60Hz refresh rate. At the time of publication, this projector could be acquired for $670USD. The specified brightness is 1100 lumens with a contrast ratio of 2000:1 allowing it to be easily viewable under common indoor office lighting. The removal of the color wheel provided us with the ability to project two gray colors of equal intensities, but with different modulation frequencies (180Hz and 360Hz) for low-perceptibility FSK transmission. We projected Gray-coded binary patterns [10] using FSK modulation at 60Hz. We used plastic fiber-optic phototransistors (Part# IF-D92 from Industrial Fiber Optics, Inc). These sensors mate with low-cost 1mm core jacketed plastic fiber optic cable allowing us to use a compact sensor package and support display surfaces of arbitrary size and geometry, Figure 4. The display surface, shown in Figure 5, is constructed using black foam board to provide a semi-rigid lightweight surface for mounting the optical sensors and the touch sensitive film. The shape of the display surface includes black tracking masks in each corner centered on each sensor. These masks help reduce the visual saliency of the tracking patterns as well as reserve physical space in the projection area that will be consumed by tracking. However, these tracking masks serve only aesthetic purposes and are not essential to the tracking system. The visible white dot in each corner is a light diffuser placed over the tip of the optical fiber to improve light absorption. The microcontroller is a Microchip PIC16F819 running at 20MHz. This microcontroller contains four 10-bit A/D converters and an analog voltage reference pin that can be used to boost sensitivity to low-voltage signals. We used a reference voltage of 0.85 volts providing an internal gain multiplier of ~6 without the need for an external amplifier. Communication to the computer was accomplished using a wired 115200baud serial connection via a Pololu Serial-to-USB adaptor. Though our prototype uses a wired connection to the host computer, the data transmission used for tracking is one-directional and requires less than 45Bytes/sec. This can be easily achieved using low-cost low-bandwidth one-way technologies, such as an infrared LED, to create fully wireless surfaces. The power consumption of our sensor package is less then 25mW during active tracking, easily provided by a small battery for several hours of continuous operation.

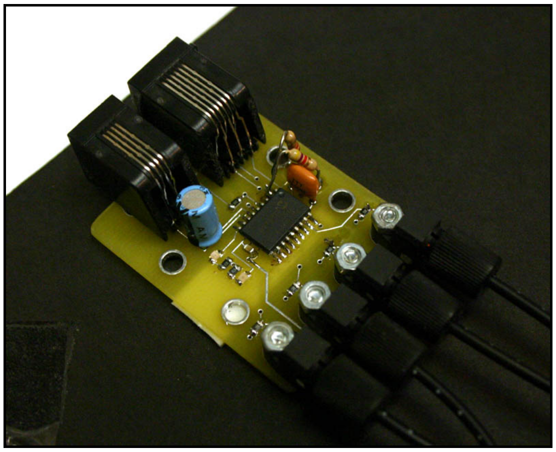

Figure 4 – Sensor board containing optical fiber connections, a PIC processor, and communication ports.

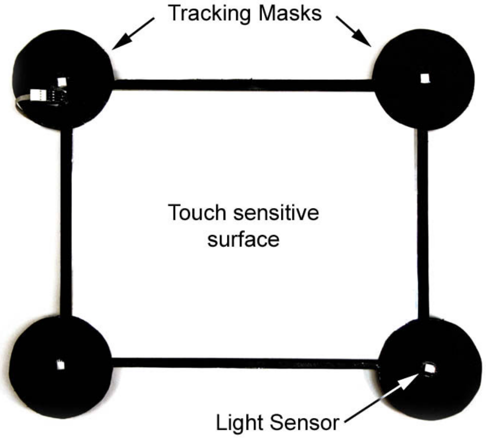

Figure 5 – Isometric view and components of the display surfaces used with our tracking system.

Sensor synchronization with the projector is accomplished using a starting 0 to 1 bit transition. This provides the sensors with a frame edge that the microcontroller uses to reset internal 60Hz timers for FSK demodulation. During continuous tracking, we perform resynchronization once every two seconds to eliminate drift in the timers resulting in >98% continuous channel utilization for tracking patterns. For the observed drift in the internal timers, this is a conservatively short sync interval and could be extended to increase channel utilization. Our full-screen discovery process uses 20 Gray-coded binary patterns and two sync bits requiring 0.367secs. For the tracking patterns, we use 5 horizontal and 5 vertical Gray-coded binary patterns to resolve a 32x32 unit grid. The pixel dimension of these patterns varied according to distance from the projector to preserve physical dimensions. The patterning time for each axis requires 83.33ms yielding an average interleaved X-Y update rate of 11.8Hz including synchronization overhead. For some applications, we linearly interpolate the tracking samples to smooth the motion of the projected content to remove the visual stuttering that results from a 12Hz update. This improves the readability and appearance of the motion-matched content, but at the cost of artificially adding lag behind the available tracking data. Our observed communication latency was approximately 60ms, which is primarily due to scheduling issues within the host PC operating system and may be substantially reduced in future implementations. Combining this with the interleaved sampling strategy, the maximum and minimum tracking latency are 185ms and 77ms respectively.

The sensor package could be manufactured for less than $10 USD in volume, making a display-only surface very inexpensive. The touch-sensitive film was purchased for $100 USD, but may be substantially cheaper in volume. In comparison, a typical price for a tablet PC at the time of publication is $1100-$1500 USD.

The demo applications described below were written in C++ using the OpenGL graphics library and executed on a Toshiba Portege M200 tablet PC running Windows XP. This tablet PC contains an NVidia GeForceGo 5200 graphics accelerator, a 1400x1050 display at a pixel density of 145ppi, a 1.7Ghz Pentium M CPU, and 512MBytes of RAM.

APPLICATIONS

Since projector based tracking unifies location tracking with the image projection technology, it greatly simplifies the implementation of systems that combine motion tracking and projected imagery. To illustrate several examples of such systems, as well as demonstrate the simplicity of this approach, we have implemented several sample applications using this technique.

Simulating Tablet-Like Displays

We can simulate hand-held displays by tracking and projecting onto a rectangular surface containing four optical sensors in each corner, shown in a Figures 1 and 6. A homography is computed from the sensor locations to prewarp projected content to fit the physical boundaries of the tablet. We also added a touch-sensitive surface allowing the user to interact with the projected content. A user can use a stylus to create free-hand drawings, take notes, or interact with a graphical user interface such as a webpage. This

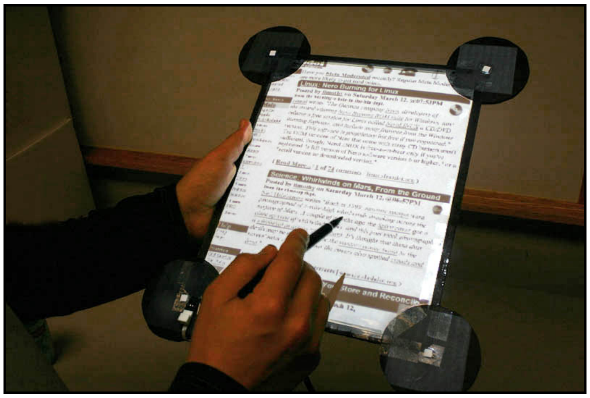

Figure 6 – Simulated tablet display with a GUI

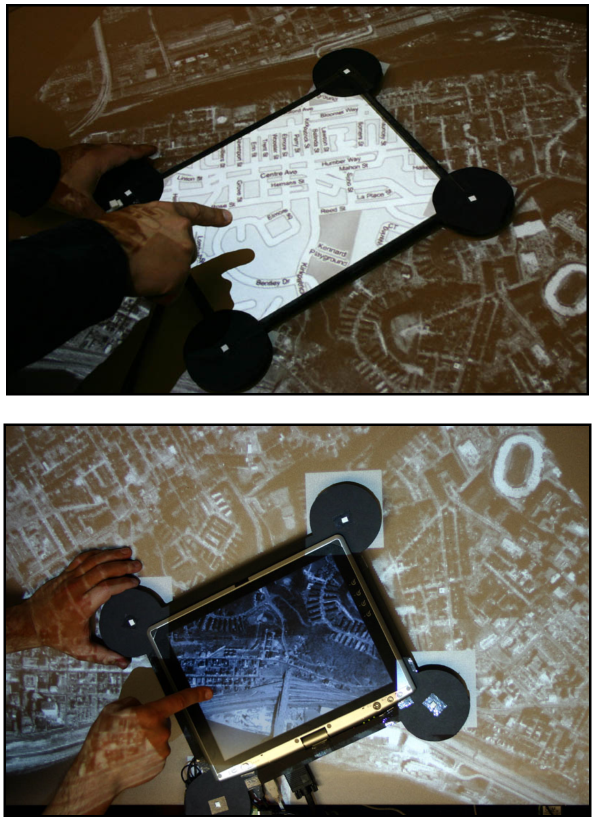

Figure 7 – *Magic Lens* (Top) and movable *Focus + Context* (Bottom)

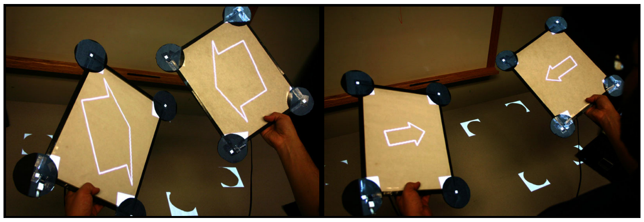

Figure 8 – Location sensitive multi-display interaction. The projected arrows reflect the direction and distance to the other surface.

effectively allows us to create functional tablet-like surfaces that are very inexpensive and weigh only slightly more than a typical clipboard. By displacing the display technology, it is possible to reduce costs by using a few ceiling mounted projectors to simulate hand-held displays in a private work environment or public space such as a museum where tablets may be given to visitors. If the surfaces are damaged, lost, or stolen, they can be easily replaced at minor expense. An environment such as a medical office might use a very large number of these surfaces to physically manage and organize patient information similar to clip boards or file folders with the added benefits of computerized tablet displays, but without the additional cost or weight. Though the performance of our current prototype is far from being able to render modern tablet PCs obsolete, improved engineering could reduce this performance gap making this a viable and practical alternative in the future.

Magic Lenses and Movable Focus Plus Context

Magic Lenses [4, 8, 19] are an elegant technique for allowing users to easily explore two-dimensional data sets containing multiple layers such as GIS map data containing aerial photographs, street data, and topography information. We can use the projection area outside the boundary of our moveable surface to display one view of the map data while the inner area provides a window into an alternative view, Figure 7. To explore a different area region of the map, the user can simply move the surface over the new area of interest. Alternatively, we can easily substitute the passive white projection surface with a high-resolution liquid crystal display (LCD) creating a moveable version of the Focus plus Context display [2]. We use four optical sensors to discover and track the corner locations of the LCD and modify the displayed content accordingly. In addition to allowing the user to choose an alternative view of the data as described before, the high-resolution display also provides a much greater level of detail than the projected image. In our implementation, we used an SVGA InFocus X1 projector and a tracked Toshiba Portege M200 tablet PC, which provided a 10:1 ratio in pixel density.

Multi-display interaction

Though our applications thus far have described using a single display surface, a projector can easily simulate more than one moveable display simultaneously, Figure 8. Each surface is tracked independently and the content for each display is warped appropriately. Additionally, because both displays are tracked using the same projector, the geometric relationship between the displays is also readily available. This information can be used to adapt the content of the two displays for display interactions such as self-orienting display spanning or intelligent transferring of objects between screens [6, 7, 15, 16].

Physical Interaction “Pucks”

We are not restricted to using rectangular surfaces containing four optical sensors. Projector based tracking is a general technique that can be used to track individual sensors, Figure 9. These sensors are packaged in black foam-board magnetic “pucks” that can be manipulated in a physical manner on a whiteboard or tabletop. These input points can then be used to define the ends of a multi-handed physical input tool such as a map measuring tool, similar to [8, 19], or to physically manipulate control elements in a planning task or simulation (e.g. a particle flow system), similar to [3].

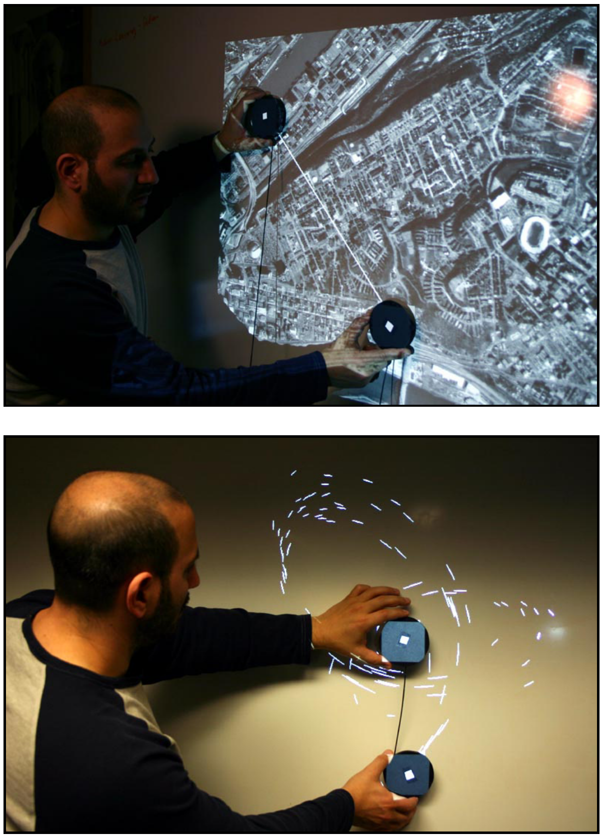

Figure 9 – Physical input “pucks” for used for a mapping tool (Top) or adjusting elements in a simulation (Bottom)

DISCUSSION AND FUTURE WORK

In this paper, we have presented a technique for using a modified commercial projector to perform motion tracking of optical sensors at interactive rates in a low-perceptibility manner. This projector based tracking technique dramatically reduces the cost and complexity of prior systems used to project motion-matched content onto moveable surfaces increasing scalability and ease of use. An incremental tracking approach does place several important limitations on the range of movement supported by the system. However, we present several strategies throughout this paper for dealing with these limitations and associated trade-offs such as adjusting pattern size and geometry, predictive motion modeling, and sensor location estimation.

Though our implementation involved removing the color wheel from the DLP projector, this is not a necessary modification to perform projector based tracking. This was a proof of concept exploration in reducing the perceptibility of the tracking patterns to minimize the visual stress as well as the user's awareness of the tracking process. Using extremely low-reflectance tracking masks may provide a sufficient reduction of the perceptibility of high-contrast patterns for some applications.

Ideally, we would like to utilize the full-capabilities of the DMD chip to project full-screen patterns at very high-speed in the infrared spectrum. This would eliminate many of the limitations related to using localized tracking patterns, preserve the color imaging capabilities of the projector, and make tracking masks unnecessary. However, these benefits come only in exchange for a significant constraint of requiring a custom DMD projector. Though we are currently pursuing this route, we are also interested in further exploring the potential uses and applications of using low-perceptibility FSK based image projection as well as novel interaction techniques enabled by projector based motion tracking.

In this work, we explored using a single projector for tracking and display purposes, which limits the physical area that a simulated display surface can be used. By using many projectors, it would be possible to expand this area of usage to encompass an entire room by dynamically moving from one projector to another as needed. Multiple projectors can also be used to reduce tracking loss resulting from occlusions using a technique similar to [18].

We are also interested in developing reusable software frameworks for applications that combine motion tracking with projected imagery to enable further exploration in this relatively new space.

REFERENCES

1. Badyopadhyay, D., Raskar, R., Fuchs, H., "Dynamic Shader Lamps: Painting on Real Objects." IEEE ACM Int. Symposium on Augmented Reality (ISAR), New York, 2001.
2. Baudisch, P., Good, N., and Stewart, P. "Focus Plus Context Screens: Combining Display Technology with Visualization Techniques." In proceedings of ACM UIST '01, Orlando, FL, November 2001, pp.31-40.
3. Ben-Joseph, E., Ishii, H., Underkoffler, J., Piper, B., and Yeung, L. "Urban Simulation and the Luminous Planning Table: Bridging the Gap between the Digital and the Tangible." In Journal of Planning in Education and Research, Volume 21, pp.195-202.
4. Bier, E., et al., "Toolglass and Magic Lenses: The Seethrough Interface." In proceedings of ACM SIGGRAPH 1993, Anaheim, CA., Aug. 1993, pp. 73-80
5. Cotting, D.; Naef, M.; Gross, M.; Fuchs, H., "Embedding Imperceptible Patterns into Projected Images for Simultaneous Acquisition and Display." Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 02-05 Nov. 2004 Page(s):100 109.
6. Hinckley, K., "Bumping Objects Together as a Semantically Rich Way of Forming Connections between Ubiquitous Devices." UbiComp 2003 formal video program.
7. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., Smith, M., "Stitching: Pen Gestures that Span Multiple Displays." In Proceedings of ACM Advanced Visual Interfaces (AVI 2004).
8. Ishii, H., and B. Ullmer. "Tangible bits: towards seamless interfaces between people, bits and atoms." In proceedings of ACM CHI'97, pp. 234-241.
9. Kjeldsen, R., Levas, A., Pinhanez, C., "Dynamically Reconfigurable Vision-Based User Interfaces" In 3rd International Conference on Vision Systems (ICVS'03). Graz, Austria. April 2003.
10. Lee, J., Dietz, P., Aminzade, D., and Hudson, S. "Automatic Projector Calibration using Embedded Light Sensors", In Proceedings of ACM UIST '04. October 2004.
11. Pinhanez, C. "The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces." In proceedings of Ubiquitous Computing 2001.
12. Raskar, R.; Beardsley, P.; van Baar, J.; Wang, Y.; Dietz, P.H.; Lee, J.; Leigh, D.; Willwacher, T., "RFIG Lamps: Interacting with a Self-Describing World via Photosensing Wireless Tags and Projectors." In proceedings of ACM SIGGRAPH. August 2004.
13. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. "The office of the future: A unified approach to image-based modeling and spatially immersive displays." In proceedings of ACM SIGGRAPH '98. ACM Press, 1998.
14. Raskar, R., Welch, G., and Low, K.L. "Shader Lamps: Animating real objects with image-based illumination." In proceedings of Eurographics Workshop on Rendering, 2001.
15. Rekimoto, J., Saitoh, M., "Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments." CHI'99, pp.378-385.
16. Streitz, N.A., Geibler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., P. Seitz, R., Steinmetz, "i-LAND: An interactive Landscape for Creativity and Innovation." In proceedings of ACM CHI'99, pp.120-127.
17. Summet, J., Sukthankar, R., "Tracking Locations of Moving Hand-held Displays Using Projected Light." In proceedings of Pervasive 2005. Munich, Germany. pp 37-46
18. Sukthankar, R., Cham, T., and Sukthankar, G. "Dynamic Shadow Elimination for Multi-Projector Displays." Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2001.
19. Ullmer, B., Ishii, H., "The metaDESK: models and prototypes for tangilble user interfaces." Proceedings of ACM UIST '97. October 1997.
20. Yotsukura,T. et al.,"Hyper Mask - Talking Head Projected onto Real Object". The Visual Computer, vol. 18(2):111-120, 2002
21. Yoder, L. "The Digital Display Technology of the Future." Texas Instrument white paper library, (www.dlp.com). Presented at Infocomm, 1997.
22. Zhang, Z., Wu, Y., Shan, Y., and Shafer, S., "Visual panel: virtual mouse, keyboard and 3d controller with an ordinary piece of paper." In Proceedings of the 2001 workshop on Perceptive User Interfaces. ACM Press, 2001.