

# TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction

Andrew D. Wilson

Microsoft Research

One Microsoft Way

Redmond, WA

awilson@microsoft.com

## ABSTRACT

A novel touch screen technology is presented. TouchLight uses simple image processing techniques to combine the output of two video cameras placed behind a semi-transparent plane in front of the user. The resulting image shows objects that are on the plane. This technique is well suited for application with a commercially available projection screen material (DNP HoloScreen) which permits projection onto a transparent sheet of acrylic plastic in normal indoor lighting conditions. The resulting touch screen display system transforms an otherwise normal sheet of acrylic plastic into a high bandwidth input/output surface suitable for gesture-based interaction. Image processing techniques are detailed, and several novel capabilities of the system are outlined.

## Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Input devices and strategies*; I.4.9 [Image Processing and Computer Vision]: Applications

## General Terms

Algorithms, Design, Human Factors

## Keywords

Computer vision, gesture recognition, computer human interaction, displays, videoconferencing

## 1. INTRODUCTION

Common touch screen technologies are limited in capability. For example, most are not able to track more than a small number of objects on the screen at a time, and typically they report only the 2D position of the object and no shape information. Partly this is due to superficial limitations of the particular hardware implementation, which in turn are driven by the emphasis on emulating pointer input for common GUI interactions. Typically, today's applications are only able to handle one 2D pointer input.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICMI'04, October 13–15, 2004, State College, Pennsylvania, USA.  
Copyright 2004 ACM 1-58113-954-3/04/0010...\$5.00.

A number of systems have recently introduced the concept of imaging touch screens, where instead of a small list of discrete points, a full *touch image* is computed, where each 'pixel' of the output image indicates the presence of an object on the touch screen's surface. The utility of the touch image thus computed has been demonstrated in gesture-based interactions for application on wall and table form factors. For example, the DiamondTouch [3] system uses horizontal and vertical rows of electrodes to sense the capacitively coupled touch of the users' hands at electrode intersections.

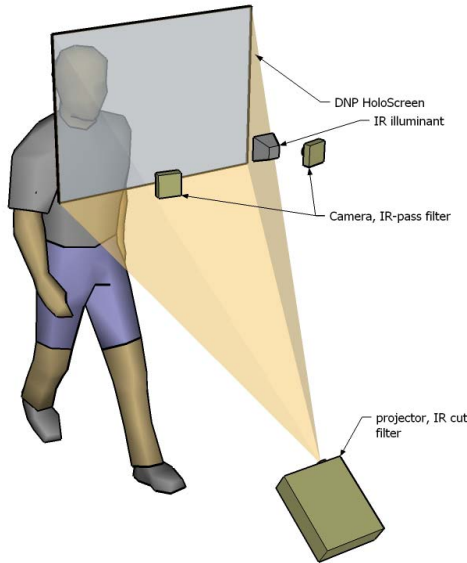
MetaDesk [13], HoloWall [9] and Designer's Outpost [8] each use video cameras and computer vision techniques to compute a touch image. These systems permit simultaneous video projection and surface sensing by using a diffusing screen material which, from the camera view, only resolves those objects that are on or very near the surface. The touch image produced by these camera-based systems reveals the appearance of the object as it is viewed from behind the surface. Application events may be triggered as the result of image processing techniques applied to the touch image. For example, the appearance or shape of an object may uniquely identify the object to the system and trigger certain application events.

In this paper we introduce the TouchLight system, which uses simple computer vision techniques to compute a touch image on a plane situated between a pair of cameras and the user (see Figures 1 and 2). We demonstrate these techniques in combination with a projection display material which permits the projection of an image onto a transparent sheet of acrylic plastic, and the simultaneous operation of the computer vision processes.

TouchLight goes beyond the previous camera-based systems; by not using a diffusing projection surface, it permits a high resolution touch image. For example, a high resolution image of a paper document may be captured using a high-resolution still camera, or one of the newer high resolution CMOS video cameras.

The absence of a diffuser also permits the cameras to see beyond the display surface, just as they would if placed behind a sheet of glass. This allows a variety of interesting capabilities such as using face recognition techniques to identify the current user, eye-to-eye video conferencing, and other processes which are typically the domain of vision-based perceptual user interfaces.

We describe the overall configuration of TouchLight, and detail the image processing techniques used to compute TouchLight's



**Figure 1 TouchLight physical configuration: DNP HoloScreen with two IR cameras and IR illuminant behind screen.**

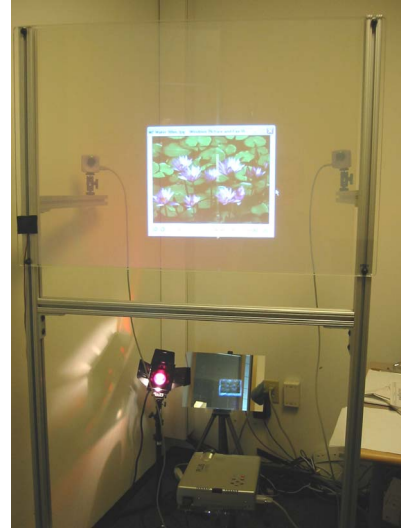
touch image. Finally, we discuss how TouchLight enables novel gesture-based interaction.

## 2. TOUCHLIGHT CONFIGURATION

The physical configuration of TouchLight is illustrated in Figure 1 and Figure 2. A pair of commonly available Firewire web cameras are mounted behind the display surface such that each camera can see all four corners of the display. The importance of the distance between the cameras is discussed later.

The DNP HoloScreen material is applied to the rear surface of the acrylic display surface. The HoloScreen is a special refractive holographic film which scatters light from a rear projector when the incident light is at a particular angle. The material is transparent to all other light, and so is suitable for applications where traditional projection display surfaces would be overwhelmed by ambient light. Typical applications include retail storefronts, where ambient light streaming through windows precludes traditional rear-projection screens. Additionally the screen is transparent in the near-infrared range. Per manufacturer’s instructions the projector is mounted such that the projected light strikes the display at an angle of about 35 degrees. In a typical vertical, eye-level installation, this configuration does not result in the user looking directly into the “hot spot” of the projector. We note that many projectors are not able to correct for the keystone distortion when the projector is mounted at this extreme angle. In our implementation, we use the NVKeystone digital keystone distortion correction utility that is available on NVidia video cards.

Experience with the HoloScreen material suggests that while the light reflected back from the rear of the screen is significantly less than the light scattered out the front, the projected image will still interfere with the image captured by any visible light-based cameras situated behind the display. In the present work we avoid difficulties with visible light reflections by conducting image-based sensing in the infrared (IR) domain. An IR illuminant is placed behind the display to illuminate the surface evenly in IR



**Figure 2 TouchLight prototype displaying a sample graphic.**

light. Any IR-cut filters in the stock camera are removed, and an IR-pass filter is applied to the lens. If necessary, an IR-cut filter may be applied to the projector. By restricting the projected light to the visible spectrum, and the sensed light to the IR spectrum, the resulting images from the camera do not include artifacts from projected light reflected backwards from the HoloScreen film.

In future work we plan to investigate the application of anti-reflection films applied to the back and also perhaps the front surface of the display to eliminate reflections from the projector. This would allow the cameras to sense visible light and perhaps eliminate the need for a separate illuminant. Later, we describe applications which benefit from visible-light based sensing.

While for our initial implementation we have chosen to mount the display vertically such that the user may stand, it is also possible to mount the display surface horizontally to make a table. In this case a “short throw” projector such as the NEC WT600 may be desirable.

Finally, a microphone is rigidly attached to the display surface to enable the simple detection of “knocking” on the display. Except for the microphone, there are no wires attached, making TouchLight more robust for public installation.

## 3. IMAGE PROCESSING

### 3.1 Introduction

The goal of TouchLight image processing is to compute an image of the objects touching the surface of the display, such as the user’s hand. Due to the transparency of the display, each camera view shows the objects on the display and objects beyond the surface of the display, including the background and the rest of the user. With two cameras, the system can determine if a given object is on the display surface or above it. TouchLight image processing acts as a filter to remove objects not on the display surface, producing a *touch image* which shows objects that are on the display surface and is blank everywhere else. A sample output image is illustrated in Figure 3d.

(a) Raw input



(b) Lens distortion correction



(c) Perspective correction



(d) Fused image



**Figure 3 TouchLight image processing steps illustrated. Images are captured in an office with normal indoor lighting: (a) raw input from both cameras, (b) input after lens distortion correction, showing display geometry during calibration, (c) input after perspective correction to rectify both views to display, and (d) fused image obtained by multiplying perspective corrected images shows only the objects that are very near the display. Hand on the left is placed flat on the display, hand on the right is slightly cupped, with tips of fingers on the display, and surface of palm above the display.**

The touch image is produced by directly combining the output of the two video cameras. Depth information may be computed by relating *binocular disparity*, the change in image position an object undergoes from one view to another view, to the depth of the object in world coordinates. In computer vision there is a long history of exploiting binocular disparity to compute the depth of every point in a scene. Such depth from stereo algorithms are typically computationally intensive, difficult to make robust, and constrain the physical arrangement of the cameras.

Often such general stereo algorithms are applied in scenarios that in the end do not require general depth maps. Here we are interested in the related but easier problem of determining what is located on a particular plane in three dimensions (the display surface) rather than the depth of everything in the scene. A related approach is taken in [14] and [2]. The algorithm detailed here runs in real time (30Hz) on a Pentium 4, operating on 640x480 images.

### 3.2 Image Rectification

The TouchLight image processing algorithm proceeds by transforming the image from the left camera  $I_{left}$  and the image from the right camera  $I_{right}$  such that in the transformed images points  $I_{left}(x, y)$  and  $I_{right}(x, y)$  refer to the same physical point on the display surface.

Secondly, this transform is such that the point  $(x, y)$  may be trivially mapped to real world dimensions (i.e., inches) on the display surface. For both criteria, it suffices to find the *homography* from each camera to the display surface, which we obtain during a manual calibration phase.

In the case of using wide angle lenses to make a compact setup, it is important to remove the effects of lens distortion imparted by wide angle lenses. We use the formulation outlined in [7]. Given the lens distortion parameters, we undistort the input image by bilinear interpolation. Sample images are shown in Figure 3b.

During a manual calibration phase, the 4 corners of the display are manually located in each view. This specifies a projective transform bringing pixels in the lens distortion corrected image to display surface coordinates. Together with the lens distortion correction, the projective transform completes the homography from camera view to display coordinates. Sample resulting images are shown in Figure 3c. We note that it is desirable to combine the lens distortion correction and projective transform into a single nonlinear transformation on the image, thus requiring only one resampling of the image. Furthermore it is straightforward to perform this entire calculation on a graphics processing unit (GPU), where the transformation is specified as a mesh.

### 3.3 Image Fusion

After rectification the same point  $(x, y)$  in both  $I_{left}$  and  $I_{right}$  refer to the same point on the display surface. Thus, if some image feature  $f$  is computed on  $I_{left}$  and  $I_{right}$ , and  $f_{left}(x, y) \neq f_{right}(x, y)$ , we may conclude that there is no object present at the point  $(x, y)$  on the display surface. The touch image mask is computed by performing such pixel-wise

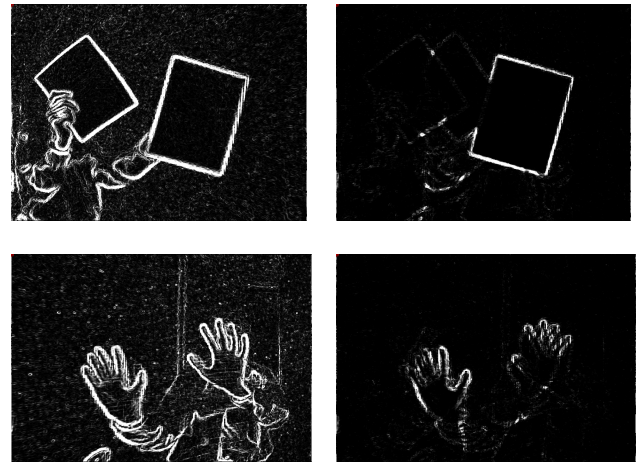
comparisons of the left and right images. This is essentially equivalent to performing standard stereo-based matching where the disparity is constrained to zero, and the rectification process serves to align image rasters.

In the case where a strong IR illuminant is available, and the goal is to identify hands and other IR reflective materials on the display surface, it may suffice to simply pixel-wise multiply the two rectified images. Regions which are bright in both images at the same location will survive multiplication. Sample resulting fused images are shown in Figure 3d. We note that it is possible to implement this image comparison as a pixel shader program running on the GPU.

As with traditional stereo computer vision techniques, it is possible to confuse the image comparison process by presenting a large uniformly textured object at some height above the display. Indeed, the height above the surface at which any bright regions are matched is related to the size of the object and to the *baseline*, the distance between the cameras. For the same size object, larger baselines result in fusion at a smaller height above the surface, consequently allowing a finer distinction as to whether an object is on the display, or just above the display.

Similarly, it is possible to arrange two distinct bright objects above the display surface such that they are erroneously fused as a single object on the surface.

More sophisticated feature matching techniques may be used to make different tradeoffs on robustness and sensitivity. For example, one possibility is to first compute the edge map of the rectified image before multiplying the two images. Figure 4 illustrates the result of applying a Sobel edge filter on the rectified images. Only edges which are present in the same location in both images will survive the multiplication. Thus, large uniform bright objects are less likely to be matched above the surface, since the edges from both views will not overlay one another. In the case of using edges, it is possible and perhaps desirable to



**Figure 4 Edge-based image fusion.** Top left: Edge extraction of one view's undistorted image (after step c in Figure 3) with sheet of paper a few inches above the display (left) and on the display (right). Top right: product of edge images. Note page above the display is not visible. Bottom: similar images for same images in Figure 3. Hand on the left is placed flat on the display, hand on the right is slightly cupped, with tips of fingers on the display, and surface of palm above the display.



reduce the baseline, resulting in better overall resolution in the rectified images due to a less extreme projective transform. The use of edge images takes advantage of the typical distribution of edges in the scene, in which the accidental alignment of two edges is unlikely.

Similarly, motion magnitude, image differences and other features and combinations of such features may be used, depending on the nature of the objects placed on the surface, the desired robustness, and the nature of subsequent image processing steps.

It should be noted that the touch plane is arbitrarily defined to coincide with the display. It is possible to configure the plane such that it lies at an arbitrary depth above the display. Furthermore, multiple such planes at various depths may be defined depending on the application. Such an arrangement may be used to implement “hover”, as used in pen-based models of interaction. The image rectification and image comparison processes do not require the physical presence of the display. In fact, it is possible to configure TouchLight to operate without the HoloScreen, in which case the “touch” interaction is performed on

an invisible plane in front of the user. In this case, it may be unnecessary to perform imaging in IR.

### 3.4 Image Normalization

A further image normalization step may be performed to remove effects due to the non-uniformity of the illumination. The current touch image may be normalized pixel-wise by

$$I_{normalized}(x, y) = \frac{I_{product}(x, y) - I_{min}(x, y)}{I_{max}(x, y) - I_{min}(x, y)}$$

where minimum and maximum images  $I_{min}$  and  $I_{max}$  may be collected by a calibration phase in which the user moves a white piece of paper over the display surface. This normalization step maps the white page to the highest allowable pixel value, corrects for the non-uniformity of the illumination, and also captures any fixed noise patterns due to IR sources and reflections in the environment.

After normalization, other image processing algorithms which are sensitive to absolute gray level values may proceed. For example, binarization and subsequent connected components algorithm, template matching and other computer vision tasks rely on uniform illumination.

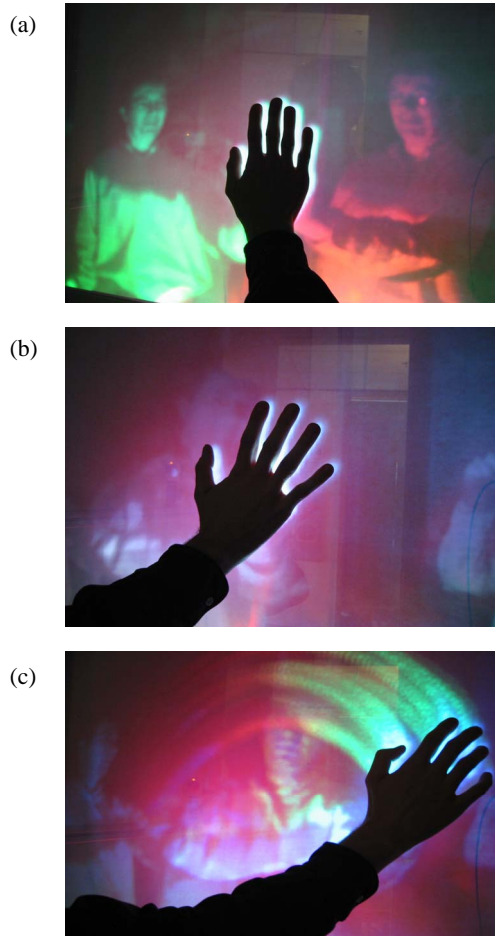
### 3.5 Touch Image Interpretation

Figure 5 shows three different visualizations of the touch image as it is projected back to the user. Figure 5a shows the user’s hand on the surface, which displays both left and right undistorted views composited together (not a simple reflection of two people in front of the display). This shows how an object fuses as it gets closer to the display. Figure 5b shows a hand on the surface, which displays the computed touch image. Note that because of the computed homography, the image of the hand indicated by bright regions is physically aligned with the hand on the screen.

Presently we have only begun exploring the possibilities in interpreting the touch image. Figure 5c shows an interactive drawing program that adds strokes derived from the touch image to a drawing image while using a cycling colormap.

Many traditional computer vision algorithms may be used to derive features relevant to an application. For example, it is straightforward to determine the centroid and moments of multiple objects on the surface, such as hands. One approach is to binarize the touch image, and compute connected components to find distinct objects on the surface (see [5]). Such techniques may also be used to find the moments of object shapes, from which dominant orientation may be determined. Further analysis such as contour analysis for the recognition of specific shapes and barcode processing are possible.

We have implemented a number of mouse emulation algorithms which rely on simple object detection and tracking. In one instance, the topmost object of size larger than some threshold is determined from a binarized version of the touch image. The position of this object determines the mouse position, while a region in the lower left corner of the display functions as a left mouse button: when the user puts their left hand on the region, this is detected as a sufficient number of bright pixels found in the region, and a left mouse button down event is generated. When the bright mass is removed, a button up event is generated. Elaborations on this have been generated, including looking for a



**Figure 5** Three different projected visualizations of TouchLight touch image: (a) left undistorted image in the green channel, right undistorted image in red channel. (b) projection of touch image illustrates alignment of touch image with physical display. (c) an interactive drawing application with decaying strokes and cycling colors.

bright mass just to the right of the tracked cursor object to detect left and right button down events when the second mass is near and far from the first, respectively.

Finally, we use a microphone rigidly attached to the display to detect “knocking” events. That is, when the user taps the display with their knuckle or hand, this is detected by finding large peaks in the digitized audio signal. This can be used to simulate clicks, generate “forward” or “next slide” events, and so on. Note that while the tap detector determines that a tap event occurred, the touch image may be used to determine *where* the event occurred. For example, a tap on the left side of the screen may generate a “previous” event, while a tap on the right a “next” event. This contrasts with the tap detector in [10].

## 4. APPLICATIONS

The unique characteristics of TouchLight lead us to speculate on some possible applications that go beyond emulating traditional touch screen technology. In the following we outline a few possibilities for future exploration.

### 4.1 Visible Light Surface Scanning

The HoloScreen display material is unique in that it supports video projection and is nearly transparent to IR and visible light. The basic TouchLight system takes advantage of this fact in the placement of the cameras behind the display. This placement provides a good view of the underside of the objects placed on the display surface. The transparency of the display surface may be exploited to create high resolution scans of documents and other objects placed on the display surface.

A high resolution still digital camera or CMOS video camera may be placed behind the display to acquire high resolution images of the objects on the display surface. This camera may capture images in the visible spectrum (no IR-pass filter). In such a configuration it may be beneficial to use the touch image computed from the IR cameras to perform detection and segmentation of objects of interest, and limit the projection of visible light onto the area of interest.

For example, an image processing algorithm may detect the presence of a letter-sized piece of paper on the display surface. The application removes any projected graphics under the presented page to enable a clear visible light view, and triggers the acquisition of a high resolution image of the display surface. The detected position, size and orientation of the page may then be used to automatically crop, straighten and reflect the high resolution scan of the document. Alternatively, the application may project an all-white graphic on the page to clearly illuminate it.

The ability to create high resolution surface scans of documents and other objects may play an important role in business and productivity oriented applications for smart surfaces such as interactive tables and smart whiteboards.

We note that related systems such as the MetaDesk, HoloWall, and Designer’s Outpost all use diffusing projection surfaces to facilitate projection and sensing algorithms. Such diffusing surfaces severely limit the ability of these systems to acquire high resolution imagery of objects on the surface.

### 4.2 Video Conferencing

The ability to place a camera directly behind the HoloScreen display, and the ability of the TouchLight system to selectively attend to objects on the surface and the scene beyond the surface may enable some interesting video conferencing scenarios.

For example, maintaining direct eye contact is impossible in today’s video conferencing systems, where the camera and the display are not co-axial. It is possible to use a half-silvered mirror to make the camera and display coaxial. This approach has been studied in the context of video conferencing systems in [1] and [6]. The use of a half-silvered mirror has the disadvantages that the brightness of the display and the acquired image is significantly reduced, the setup requires large amounts of space in front of the display, and finally, the configuration imposes restrictions on viewing angle.

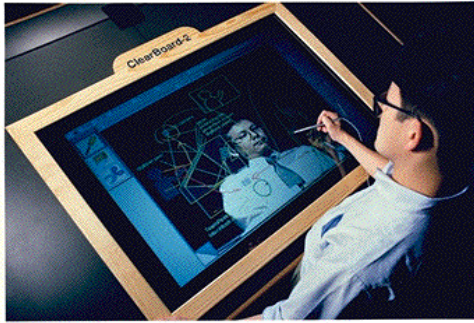
An eye-to-eye video conferencing system may be constructed by placing a video camera directly behind the TouchLight display surface. The chief difficulty in constructing such a system is that if the camera used is acquiring IR images so as to avoid artifacts from the projected image, the resulting imagery may not be satisfactory for presentation back to the user. Alternatively, if the camera acquires visible light images, then the presentation must be carefully crafted so that the acquired image does not include any light scattered back from the rear of the display surface. The application of an anti-reflective film on the front and rear of the HoloScreen material may eliminate the back reflection. We also note that it is theoretically possible to use image processing techniques to remove artifacts due to the projection since the system has access to the projected image and the homography from the camera to the display surface is known.

The ability to place a camera behind the screen may have uses beyond eye-to-eye video conferencing. Even with the grayscale IR image returned by TouchLight, it will be possible to determine who is interacting with the display surface by face recognition techniques, determine whether they are looking at the display and possibly even where on the display the user is looking. Such capabilities may be relevant in multi-user and collaborative scenarios. Perhaps uncomfortably, such analysis can be conducted with the cameras completely concealed behind the display surface.

A number of research projects have explored video conferencing displays which are loosely modeled as panes of glass in which two non co-located users are able to see each other manipulate objects rendered on the display. ClearBoard [6] is an early example (see Figure 6). We foresee the applicability of this window metaphor in using TouchLight in video conferencing scenarios. Note that the ability to create high resolution scans outlined in the previous section may be especially valuable in this scenario.

### 4.3 Minority Report Interfaces

Movies such as *Minority Report* and *The Matrix Reloaded* have popularized the idea of gesture and direct manipulation-based interfaces involving transparent displays. Of the hundreds of people that have seen TouchLight demos, roughly half made unsolicited comparisons of TouchLight to the interaction systems shown in these two movies. The value of the transparency of the displays used in these future visions is debatable. Clearly, the transparency taps into the public’s fascination with holograms, but more mundanely it creates the opportunity for filmmakers to



**Figure 6** ClearBoard-2 illustrates shared drawing surface and eye contact between remote participants. Image courtesy Hiroshi Ishii and NTT Human Interface Laboratories.

cleanly put the interaction system and the actor's face in the same shot.

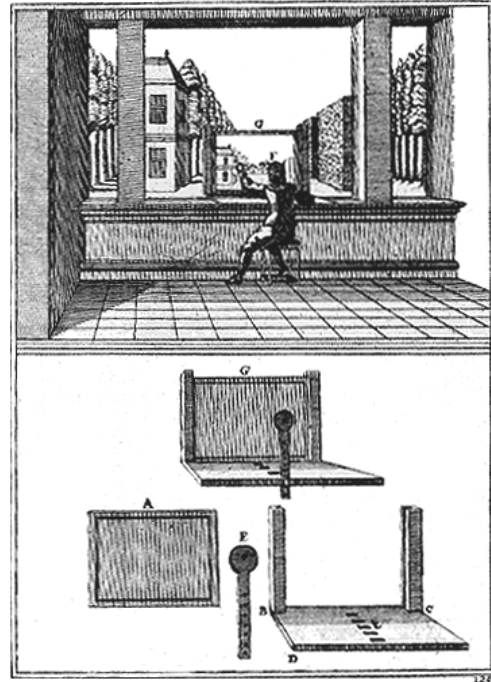
Several research projects, however, are taking seriously the gesture-based manipulation of onscreen objects [15] [11] in the style of direct manipulation. For certain classes of interaction, this style of interaction seems to be more natural than the traditional WIMP (windows, icons, menus, pointer) interface. For example, sorting through a stack of photos may be more easily conducted in a direct manipulation framework that allows the use of multiple hands, taking advantage of our own abilities to sort objects into groups or piles [12]. Objects may be rotated in a way that mimics the rotation of a physical piece of paper on a desk. Certain collaborative exercises may benefit from direct manipulation, where each user may easily comprehend the other users' actions. We suspect that direct manipulation frameworks are more readily picked up by novice users, and therefore are suited to quick serendipitous interactions, perhaps at public kiosks, or in short face to face, collaborative meetings. In these situations the overhead in acquiring an input modality may mean the difference between conducting an interaction or not.

#### 4.4 Augmented Reality and Spatial Displays

With the ability to project on a transparent display, TouchLight enables scenarios where projected graphics are overlaid onto imagery from the real world. The application of the HoloScreen material for an augmented reality application is explored in [4], which describes a boom-mounted and instrumented screen and projector system used to overlay graphics onto the real world beyond the screen.

TouchLight raises new possibilities for augmented reality and spatial displays. For example, imagine a retail environment installation where customers are invited to try on virtual articles of clothing while looking at themselves in a TouchLight "mirror". In this scenario, a camera may be placed to synthesize the view the customer would have if they looked into a real mirror. A computer graphics system would composite the clothing onto the view in real time as the customer moves, while TouchLight interaction may allow the user to select various articles of clothing on their mirror image, or interact with buttons alongside their image.

With the touch sensitive capabilities of TouchLight, scenarios inspired by the concept of Alberti's Veil or Leonardo's Window are possible. Alberti's Veil is a technique still used to teach



**Figure 7** A typical device after Leonardo's window. Such devices were used to teach perspective in architecture. From P. Le Dubreuil, *La Perspective Pratique*, 1649

perspective whereby a scene projected onto a window is traced, with the artist maintaining a stationary viewpoint (see Figure 7). With TouchLight, an artist may trace or modify a visual scene, and with computer vision techniques it is possible to track the face of the user and perhaps detect gaze direction to correct for parallax from the user's point of view to the display in aligning projected graphics with the real world. Many spatial display systems are based on the ability to track the user's face and eyes.

#### 5. CONCLUSION

A novel interactive surface and touch screen technology is presented. TouchLight uses two cameras in combination with a commercially available projection screen technology which allows projection onto an otherwise transparent surface. This arrangement allows for certain novel applications and flexibility which go beyond previous related technologies.

We have presented image processing techniques to produce a touch image useful for many gesture-based and perceptual computing scenarios. A number of applications which take advantage of the unique characteristics of TouchLight have been suggested; we hope to explore some of these in the future.

#### 6. REFERENCES

1. Buxton, B., T. Moran, EuroPARC's Integrated Interactive Intermedia Facility (IIIF): Early Experiences. in *IFIP WG8.4 Conference on Multi-User Interfaces and Applications*, (1990), 11-34.
2. de la Hammette, P., P. Lukowicz, G. Tröster, T. Svoboda, Fingermouse: A Wearable Hand Tracking System. in *Ubicomp 2003: Ubiquitous Computing*, (2002).

3. Dietz, P.H., D. L. Leigh, DiamondTouch: A Multi-User Touch Technology. in *ACM Symposium on User Interface Software and Technology (UIST)*, (2001), 219-226.
4. Ferscha A., M.K., DigiScope: An Invisible Worlds Window. in *Adjunct Proceedings, The Fifth International Conference on Ubiquitous Computing*, (Seattle, 2003), 261-264.
5. Horn, B.K.P. *Robot Vision*. MIT Press, Cambridge, MA, 1986.
6. Ishii, H., M. Kobayashi, ClearBoard: A Seamless Media for Shared Drawing and Conversation with Eye-Contact. in *Conference on Human Factors in Computing Systems (CHI)*, (1992), 525-532.
7. Kang, S.B. Radial Distortion Snakes. *IEICE Transactions on Information and Systems*, E84-D (12). 1603-1611.
8. Klemmer, S.R., M. W. Newman, R. Farrell, M. Bilezikjian, J. A. Landay, The Designer's Output: A Tangible Interface for Collaborative Web Site Design. in *ACM Symposium on User Interface Software and Technology*, (2001), 1-10.
9. Matsushita, N., J. Rekimoto, HoloWall: Designing a Finger, Hand, Body and Object Sensitive Wall. in *ACM Symposium on User Interface Software and Technology (UIST)*, (1997).
10. Paradiso, J.A., C. K. Leo, N. Checka, K. Hsiao, Passive Acoustic Knock Tracking for Interactive Windows. in *ACM Conference on Human Factors in Computing: CHI 2002*, (2002), 732-733.
11. Ringel, M., K. Ryall, C. Shen, C. Forlines, F. Vernier, Release, Rotate, Reorient, Resize: Fluid Techniques for Document Sharing on Multi-User Interactive Tables. in *Short Paper, ACM Conference on Human Factors in Computing Systems*, (2004).
12. Shen, C., F.D. Vernier, C. Forlines, M. Ringel, DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction. in *ACM Conference on Human Factors in Computing Systems (CHI)*, (2004).
13. Ullmer, B., H. Ishii, The metaDESK: Models and Prototypes for Tangible User Interfaces. in *ACM Symposium on User Interface Software and Technology*, (1997), 223-232.
14. Wren, C.R., Y. A. Ivanov, Volumetric Operations with Surface Margins. in *Computer Vision and Pattern Recognition: Technical Sketches*, (2001).
15. Wu, M., R. Balakrishnan, Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. in *ACM Symposium on User Interface Software and Technology*, (2003), 193-202.