

# Vision-based Projected Tabletop Interface for Finger Interactions

Peng Song, Stefan Winkler, Syed Omer Gilani, and ZhiYing Zhou

Interactive Multimedia Lab  
Department of Electrical and Computer Engineering  
National University of Singapore, Singapore 117576  
`{psong,winkler,omer,elezzy}@nus.edu.sg`

**Abstract.** We designed and implemented a vision-based projected tabletop interface for finger interaction. The system offers a simple and quick setup and economic design. The projection onto the tabletop provides more comfortable and direct viewing for users, and more natural, intuitive yet flexible interaction than classical or tangible interfaces. Homography calibration techniques are used to provide geometrically compensated projections on the tabletop. A robust finger tracking algorithm is proposed to enable accurate and efficient interactions using this interface. Two applications have been implemented based on this interface.

**Key words:** Projector-camera systems, projector-based display, interface design, tabletop interface, homography calibration, finger tracking, augmented reality, bare-hand interface.

## 1 Introduction

As projectors are becoming smaller and more affordable, they are used in more and more applications to replace traditional monitor displays because of their high scalability, larger display size and setup flexibility. Projection displays are no longer limited to traditional entertainment uses such as showing movies, playing video games, etc. They are now also being used for education [1], visualization [2], simulation [3], design [4] and interaction [5] applications.

The input to these applications is traditionally mouse-and-keyboard based, which is unnatural for users and limits their interactions and flexibility. Tangible interfaces have also been used in some projected environments. By holding some physical objects in hand, users feel comfortable manipulating the objects for interaction. Sensetable [6] uses a projected interface for visualization and design. Physical objects with embedded sensors can be held by users for movements to represent corresponding interactions. The Flatland system [7] projects onto a normal whiteboard, and interactions are based on the interpretation of strokes by the stylus on the whiteboard. More recently, Escritoire [8] uses special pens with embedded sensors to enable user interaction between a user and a projected tabletop. These applications are all based on manipulating tangible objects including pens for interactions. However, the flexibility can be greatly improved

through using only hands and fingers. Barehand interfaces enjoy higher flexibility and more natural interaction than tangible interfaces. DiamondTouch [9] and SmartSkin [10] both rely on the users' hands for interaction, however, a grid of wired sensors needs to be embedded in the tabletop to enable the barehand interaction. Such special hardware and materials are expensive and not readily accessible to everyone.

As cameras can be used to track the hand and finger movements through computer vision approaches, we propose a vision-based projected tabletop interface with finger tracking. Webcams are very cheap and non-intrusive sensors, therefore the tracking of fingers by a webcam not only provides more flexible, natural and intuitive interaction possibilities, but also offers an economic and practical way of interaction for the user and the computer.

The design of this interface will be presented in section 2, followed by a discussion on homography calibration in section 3. In section 4, a robust finger tracking algorithm will be proposed. Two applications based on this interface, and the finger tracking results will be reported in section 5.

## 2 Interface Design

The projector and the camera are mounted on tripods with an overlapping view of the tabletop. The projector is projecting from the upper left onto the tabletop, and the camera is observing the projected area from the upper right. The projector and the camera can be randomly put in the above mentioned manner without any calibration/measurement. In this setup, the camera can see the hand and its shadow under the projector illumination during the interaction.

As shown in Figure 1, the camera is overlooking the projected content on the tabletop and the hand as well. The geometric distortion caused by the obliquely positioned projector can be compensated through homography transformation on the image before projection (see section 3). The finger and its shadow can be tracked using the camera input, and tracks of the finger can be projected onto the tabletop in a handwriting recognition system, or used to indicate the direction and location a document is dragged to simulating typical desk operations.

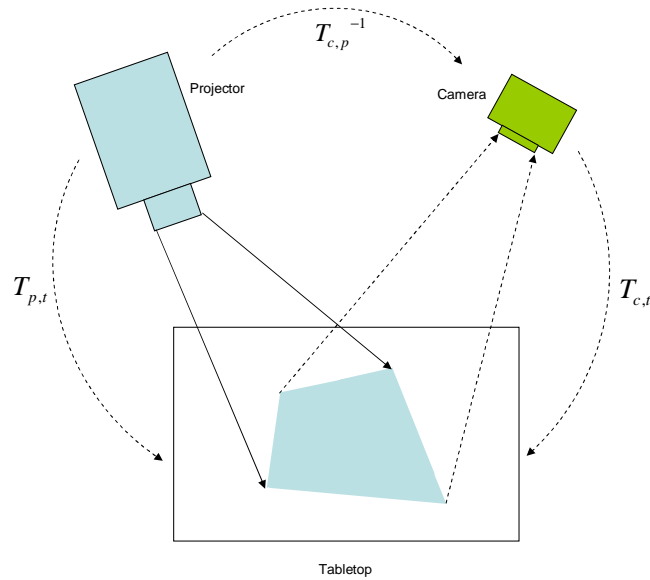
Our system only requires a commodity projector with an off-the-shelf webcam. Users need no other equipment or devices for interaction, hence the system provides direct, natural and intuitive interactions between the user and the computer.

## 3 Homography Calibration

In our setup, the projector and the camera can be casually set up at oblique angles for minimizing the occlusion by the human hand in the projection area. As a result, images sent to the projector need to be geometrically compensated for the projective distortions, as shown in Figure 2, in order to achieve an undistorted view from the user's perspective.



**Fig. 1.** System setup with the projector and the camera overlooking the tabletop. The projector and the camera can be casually and obliquely positioned, as long as the camera can see the projection area.



**Fig. 2.** Homography calibration. Due to misalignment of projector and projection surface (tabletop), the rectangular display appears as a distorted quadrilateral in the center of the tabletop. This can be compensated by pre-warping using the homographies shown.

As the distortion is predominantly the result of perspective projection, pre-warping the image via a homography can be used to correct for the distortion. The homographies between projector, camera, and the tabletop need to be calibrated prior to pre-warping. Sukthankar *et al.* [1, 11] proposed an automatic keystone correction method for the above setup. A simple solution using four feature point correspondences is shown in Figure 2. If a white rectangle is projected onto the tabletop against a high-contrast background, the four corners of the projected quadrilateral in the camera image can be computed. With the coordinates of the corners known in the projector reference frame, the projector-camera homography  $\mathbf{T}_{cp}$  can be recovered. The corners of this quadrilateral completely specify the camera-tabletop homography  $\mathbf{T}_{ct}$ . The homography between the projector and the tabletop  $\mathbf{T}_{pt}$  can be recovered from  $\mathbf{T}_{pt} = \mathbf{T}_{cp}^{-1}\mathbf{T}_{ct}$ . By applying the inverse transform  $\mathbf{T}_{pt}^{-1}$  to pre-warp the original image, a corrected image can be displayed on the tabletop.

## 4 Fingertip Tracking

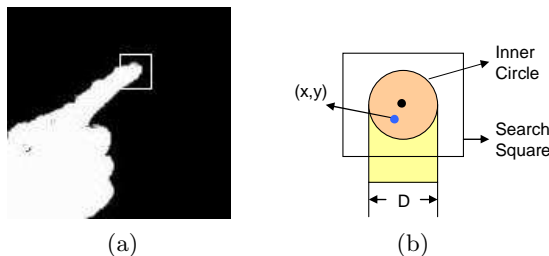
Because our projected tabletop interface provides finger interaction, the finger of the user needs to be tracked in the camera view such that interactions can be enabled accordingly. There are many vision-based techniques that can track fingers. However, there are many constraints on these methods: methods based on color segmentation [12] need users to wear colored gloves for efficient detection; wavelet-based methods [13] are computationally expensive and non real-time; contour based methods [14] work only on restricted backgrounds; infrared segmentation based methods [15, 16] require expensive infrared cameras; correlation-based methods [17] require an explicit setup stage before the tracking starts; the blob-model based method [18] imposes restrictions on the maximum speed of hand movements. In order to provide robust real-time hand and finger tracking in the presence of rapid hand movements and without the need of initial setup stage, we propose an improved finger tracking based on Hardenberg’s fingertip shape detection method [19] with more robustness and the ability to detect the event of a finger touching the surface.

### 4.1 Fingertip Shape Detection

Hardenberg [19] proposed a finger tracking algorithm using a single camera. With its smart image differencing, the user’s hand can be easily distinguished from the static background. Fingertips are then needed to be detected for interaction purposes.

In a difference image, the hand is represented in filled pixels, while the background pixels are all unfilled, as shown in Figure 3(a). If a square box is shown around a fingertip, as in Figure 3(b), the fingertip shape is formed by circles of linked pixels around location  $(x, y)$ , a long chain of non-filled pixels, and a short chain of filled pixels.

In order to identify the fingertip shape around pixel location  $(x, y)$ , 3 criteria have to be satisfied:



**Fig. 3.** Fingertip Shape Detection. (a) Difference image shows the fingertip shape of filled pixels on a background of non-filled pixels. (b) The fingertip at position  $(x, y)$  can be detected by searching in a square box (see text).

1. In the close neighborhood of position  $(x, y)$ , there have to be enough filled pixels within the searching square;
2. The number of filled pixels has to be less than that of the non-filled pixels within the searching square;
3. The filled pixels within the searching square have to be connected in a chain.

The width of the chain of filled pixels can be expressed as  $D$  and is the diameter of the identified finger.  $D$  needs to be preset/adjusted in order to detect fingers of different diameters in the camera view.

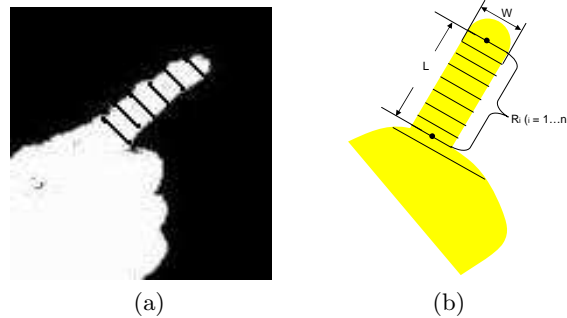
This method works well under normal lighting conditions. However, under the strong projector illumination in our setup, a lot of false fingertip detections will appear because of the failure of smart differencing in such conditions. In order to detect the fingertips more robustly, a finger shape detection method is proposed in the following section.

## 4.2 Finger Shape Detection

The above detection method may produce false detections on the finger end in connection with one's palm, as shown in Figure 4(a), because of its similar shape. However, these false detections can be eliminated if the shape of the finger is taken into consideration. As a fingertip always has a long chain of filled pixels connected to the palm, and the width of the chain of filled pixels will be greatly changed only at the connection from the finger to the palm, a more robust finger detection algorithm based on the shape of the finger is proposed as follows.

When a fingertip is detected from the method used in section 4.1, record the center of the fingertip, move further along the direction of the chains of filled pixels. As shown in Figure 4(b), the width  $W_i$  ( $i = 1 \dots n$ ) of the  $i^{th}$  row of filled pixels ( $R_i$  ( $i = 1 \dots n$ )) orthogonal to the direction of the detected chain of filled pixels is computed. If the width of row  $W_{i+1}$  is comparable with  $W_i$ , move along the direction of the chain of filled pixels, until the width increase between row  $W_n$  and  $W_{n-1}$  increases dramatically.

The length of the potential finger  $L$  can be derived by computing the distance between the center pixel of row  $R_1$  and that of row  $R_{n-1}$ . A finger is confirmed



**Fig. 4.** Finger Shape Detection. (a) Difference image showing the finger shape of a long chain of filled pixels connected with the palm’s chain of filled pixels. The finger is labeled by rows of black pixels orthogonal to the direction of the chain of filled pixels detected in Hardenberg’s method [19]. (b) A finger can be represented by a long chain of filled pixels. Along the finger, the width  $W$  of chain of filled pixels orthogonal to the direction of the detected chain of filled pixels will change dramatically at the connection of a finger to its palm.

if  $L$  is sufficiently long, since false detected fingertips normally do not have long fingers attached.

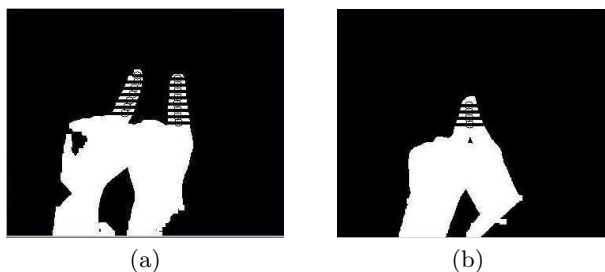
Through employing the finger information, this algorithm effectively eliminates the false fingertip detections based on the fingertip shape information.

### 4.3 Touch-table Detection

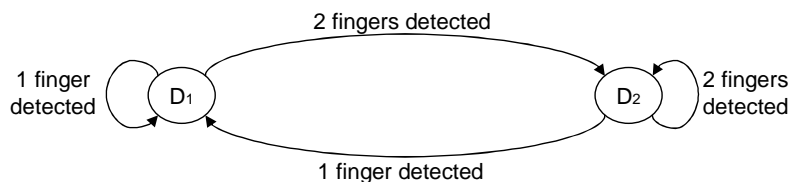
Since only a single camera is used for detection, no depth or touch information is available to identify if the finger is touching the tabletop. However, under projector illumination in our setup, prominent finger shadows are observed, which are also detected as fingers by the above algorithm. Therefore, when two detected fingertips (the real finger and its shadow) merge into one, we can assume that the finger touches the tabletop.

In order to keep tracking the finger merged with its shadow while it is moving on the tabletop as shown in Figure 5(b), the diameter  $D$  of the finger detection algorithm has to be adjusted. If a finger moving in the air is to be detected, the diameter of the fingertip is set to  $D_2$ . Likewise, if a finger touching the tabletop is to be detected, the diameter of the fingertip is set to  $D_1$  where  $D_1 > D_2$ . The diameters  $D_2$  and  $D_1$  can be pre-determined accordingly. However, in order to automatically switch between these two diameters when a finger is touching and leaving the tabletop, a two-state system is proposed based on the assumption that there is only one finger used in the interaction.

As shown in Figure 6, an initial state is set to 2 fingers tracked with diameter  $D_2$ , as the user would first move his hand in the air before touching the tabletop. The state of each frame can be determined using the fingertip diameter  $D$  in the previous frame, and then  $D$  may be adjusted according to the state diagram.



**Fig. 5.** Touch Table Event Detection. (a) Finger moving in the air; both its shadow and itself are detected as separate fingers. (b) Finger touching the table; it merges with its shadow, resulting in the detection of a single wide finger.



**Fig. 6.** State diagram for finger diameter switching.

## 5 Applications and Results

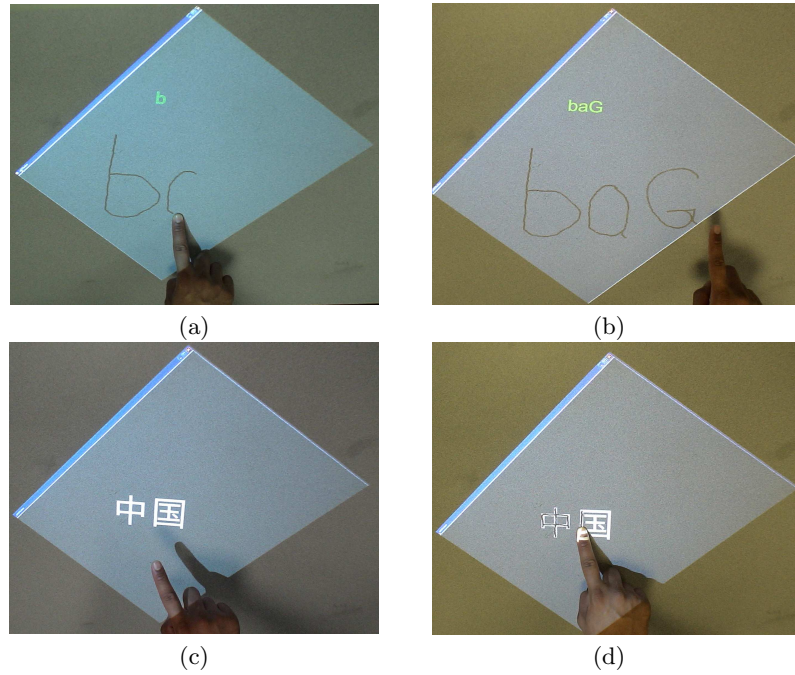
Our system comprises an Optoma EP-739H DLP portable projector, a Logitech webcam and a Dell Optiplex GX620 Desktop PC. Two applications are tested on this interface and shown below.

### 5.1 Handwriting Recognition and Template Writing

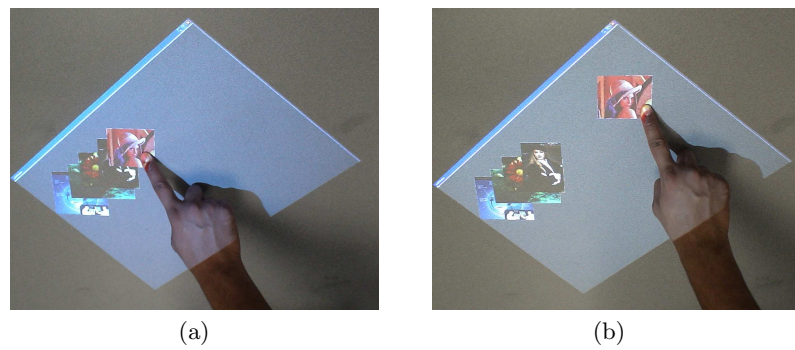
Through tracking the forefinger of a user on the tabletop, the stroke information can be projected onto the tabletop after geometric compensation. Based on Microsoft Windows XP Tablet PC Edition 2005 Recognizer Pack [20], recognized letters can be shown on the upper-right corner in the projection area (shown in Figure 7(a,b)). In addition, to train kids to learn how to write, colored templates can be projected onto the tabletop, and a user can write on the template while his writing is shown in projected strokes (shown in Figure 7(c,d)).

### 5.2 Desktop Finger Mouse

Because the finger of a user can be tracked both when it is moving in the air and when moving on the tabletop, a few mouse functions can be replaced by finger touching the table, moving on the tabletop, etc. Shown in Figure 8 is the action of a user selecting a file (a) and drags it to another location in the projection area (b).



**Fig. 7.** Screenshots of Handwriting Recognition System and Template Writing. A user completing the writing of “baG” using his finger and recognized by our system (green printed words) is shown in (a,b); two Chinese characters are projected in (c) and the user is using his hand to following the strokes to practice Chinese calligraphy (d).



**Fig. 8.** Desktop Finger Mouse. A user is using his finger as a mouse to “select” a picture on the left (shown in (a)) and “drag” it to the right part of the display (shown in (b)).



### 5.3 Fingertip Detection Results

Our improved finger detection algorithm proposed in section 4 is more robust. In table 1, the finger tracking accuracy rates of our method is shown in comparison with Hardenberg’s Method [19]. Among 180 sampled frames, the average correct detection percentage of our method (88.9%) is much higher than that of their method (46.1%). Among these frames, the finger can be either correctly detected, or not detected at all, or detected with additional 1 false detection, or detected with additional 2 false detections. The number of frames corresponding to each type of detection is shown in table 1 below.

**Table 1.** Finger Tracking Accuracy Rates

|                     | Types              | 180 Frames | Average Accuracy |
|---------------------|--------------------|------------|------------------|
| Hargenberg’s method | correct            | 83         | 46.1%            |
|                     | 1 false detection  | 71         |                  |
|                     | 2 false detections | 26         |                  |
|                     | no detection       | 0          |                  |
| Our method          | correct            | 160        | 88.9%            |
|                     | no detection       | 20         |                  |

## 6 Conclusions and Future Work

We have designed and implemented a vision-based projected tabletop interface for finger interactions. Our interface provides more direct viewing, natural and intuitive interactions than HMD’s or monitor-based interfaces. The system setup is simple, easy, quick and economic since no extra special devices are needed. Homography calibration is used to compensate for distortions caused by the oblique projection. A fast finger tracking algorithm is proposed to enable robust tracking of fingers for interaction.

We are now conducting a usability study to evaluate the interface from a user perspective. Furthermore, due to the oblique projection, there are issues such as projection color imbalance and out-of-focus projector blur that need to be investigated.

## References

1. Sukthankar, R., Stockton, R., Mullin, M.: Smarter Presentations: Exploiting homography in camera-projector systems. In: Proc. International Conference on Computer Vision, Vancouver, Canada (2001) 247–253
2. Chen, H., Wallace, G., Gupta, A., Li, K., Funkhouser, T., Cook, P.: Experiences with scalability of display walls. In: Proc. Immersive Projection Technology Symposium (IPT), Orlando, FL (2002)

3. Czernuszenko, M., Pape, D., Sandin, D., DeFanti, T., Dawe, L., Brown, M.: The ImmersaDesk and InfinityWall projection-based virtual reality displays. *Computer Graphics* (1997) 46–49
4. Buxton, W., Fitzmaurice, G., Balakrishnan, R., Kurtenbach, G.: Large displays in automotive design. *IEEE Computer Graphics and Applications* **20**(4) (2000) 68–75
5. Song, P., Winkler, S., Tedjokusumo, J.: A tangible game interface using projector-camera systems. In: *Proc. HCI International Conference, Beijing, China* (2007)
6. Patten, J., Ishii, H., Pangaro, G.: Sensetable: A wireless object tracking platform for tangible user interfaces. In: *Proc. CHI, Conference on Human Factors in Computing Systems, Seattle, Washington, USA* (2001)
7. Mynatt, E.D., Igarashi, T., Edwards, W.K.: Flatland: New dimensions in office whiteboards. In: *Proc. CHI'99, Pittsburgh, PA., USA* (1999)
8. Ashdown, M., Robinson, P.: Escritoire: A personal projected display. *IEEE Multimedia Magazine* **12**(1) (2005) 34–42
9. Leigh, D., Dietz, P.: DiamondTouch characteristics and capabilities. In: *UbiComp 2002 Workshop on Collaboration with Interactive Tables and Walls, Göteborg, Sweden* (2002)
10. Rekimoto, J.: SmartSkin: An infrastructure for freehand manipulation on interactive surfaces. In: *Proc. CHI'02, Göteborg, Sweden* (2002) 113–120
11. Sukthankar, R., Stockton, R., Mullin, M.: Automatic keystone correction for camera-assisted presentation interfaces. In: *Proc. International Conference on Multimodal Interfaces, Beijing, China* (2000) 607–614
12. Lien, C., Huang, C.: Model-based articulated hand motion tracking for gesture recognition. *Image and Vision Computing* **16**(2) (1998) 121–134
13. Triesch, J., Malsburg, C.: Robust classification of hand postures against complex background. In: *Proc. International Conference On Automatic Face and Gesture Recognition, Killington* (1996)
14. Segen, J.: Gesture VR: Vision-based 3D hand interface for spatial interaction. In: *Proc. ACM Multimedia Conference, Bristol, UK* (1998)
15. Rehg, J., Kanade, T.: Digiteyes: Vision-based 3D human hand tracking. In: *Technical Report CMU-CS-93-220, School of Computer Science, Carnegie Mellon University* (1993)
16. Sato, Y., Kobayashi, Y., Koike, H.: Fast tracking of hands and fingertips in infrared images for augmented desk interface. In: *Proc. International Conference on Automatic Face and Gesture Recognition, Grenoble, France* (2000)
17. Crowley, J., Bérard, F., Coutaz, J.: Finger tracking as an input device for augmented reality. In: *Proc. International Conference on Automatic Face and Gesture Recognition, Zürich, Switzerland* (1995)
18. Laptev, I., Lindeberg, T.: Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features. In: *Technical Report ISRN KTH/NA/P-00/12-SE, The Royal Institute of Technology (KTH), Stockholm, Sweden* (2000)
19. Hardenberg, C., Brard, F.: Bare-hand human computer interaction. In: *Proc. Perceptual User Interfaces, Orlando, Florida, USA* (2001)
20. Microsoft Corporation: Microsoft Windows XP Tablet PC Edition 2005 Recognizer Pack. <http://www.microsoft.com/downloads/details.aspx?familyid=080184dd-5e92-4464-b907-10762e9f918b&displaylang=en>