

# A New Method for Multi-finger Detection Using a Regular Diffuser

Li-wei Chan, Yi-fan Chuang, Yi-wei Chia, Yi-ping Hung, and Jane Hsu

Graduate Institute of Networking and Multimedia  
Department of Computer Science and Information Engineering  
National Taiwan University  
hung@csie.ntu.edu.tw

**Abstract.** In this paper, we developed a fingertip finding algorithm working with a regular diffuser. The proposed algorithm works on images captured by infra-red cameras, settled on one side of the diffuser, observing human gestures taken place on the other side. With diffusion characteristics of the diffuser, we can separate finger-touch from palm-hover events when the user interacts with the diffuser. This paper contributes on: Firstly, the technique works with a regular diffuser, infra-red camera coupled with an infra-red illuminator, which is easy to deploy and cost effective. Secondly, the proposed algorithm is designed to be robust for casually illuminated surface. Lastly, with diffusion characteristics of the diffuser, we can detect finger-touch and palm-hover events, which is useful for natural user interface design. We have deployed the algorithm on a rear-projection multi-resolution tabletop, called I-M-Top. A video retrieval application using the two events on design of UIs is implemented to show its intuitiveness on the tabletop system.

**Keywords:** Multi-Finger Detection, Intuitive Interaction.

## 1 Introduction

Work with displays in life size, direct manipulation using bare hand gesture is more nature and more like how people interact with the real world. Recently, researchers have been studying on design of user interface on digital surfaces such as interactive wall and tabletop. Among which, many works are based on the visual hand gesture recognition techniques. Obviously, for applications that need for complex operations and rich interactions, a finger-level gesture recognition technique is necessary.

In this paper, a fingertip finding algorithm applied with a regular diffuser is proposed. The algorithm works on images captured by infra-red cameras, settled on one side of the diffuser, observing human gestures taken place on the other side. On user's hands approaching the diffuser lighted by the infra-red illuminator, the camera observes reflection left by the hands to do further recognition. The installation is quiet simple, cost-effective and flexible to deploy with digital surface achieved by either front or rear projection.

The proposed algorithm is able to separate fingertip interactions into two events: finger-touch (figure 1a-1c) and palm-hover (finger 1d) events. Finger-touch event

refers to the fingertips are contacting the surface and palm-hover event refers to the user's palms are hovering over the display surface. The two kinds of events can be used in designing user interfaces reacting accordingly. For example, when the user's palms approach the display surface, the content beneath the palms can change accordingly so as to give more specific selections for upcoming fingertip actions.

## 2 Related Work

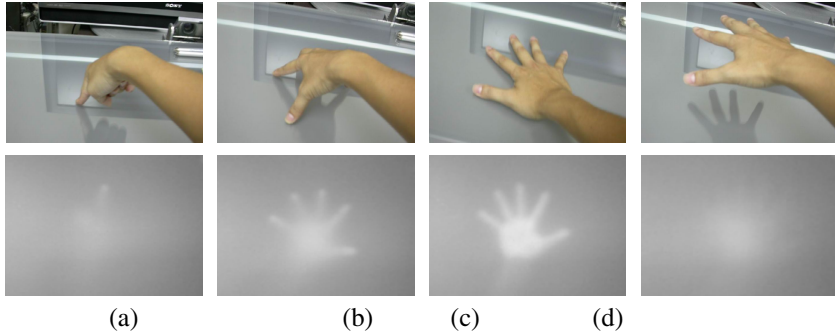
Previous researches working on finger tracking on interactive wall or tabletop systems using computer vision techniques have exploited ways of installations. These works can differ in how cameras installed to observe human hand gestures. In one kind of installations, the camera is settled to be able to directly observe the hand gestures [2-5, 7, 8, and 10], so clear hand shape segmentations are expected for analysis. In [2], their work presented a robust finger tracking algorithm by using single camera, but they cannot distinguish whether the user's fingers touch the surface. In [10], the work proposed analyzing finger shadows shown on the surface to detect finger-touch events. With the use of two cameras, these works [4, 5, and 7] can track fingers and detect finger-touch event on the surface.

Another installation is to have cameras installed behind a transparent sheet, and to observe hand gestures through the surface. In [1], they use an infra-camera and infra-red illuminator placed on one side of a diffuser. The installation in their work is similar to ours, but they only provide palm-level recognition, so only simple operations were presented. Working with two cameras, TouchLight [6] proposed an effective fingertip finding approach by using stereo information, but a special diffuser, HoloScreen, is required. FTIR[9] is another work with excellent performance to track multiple fingertips when the users place hands on the surface, but hovering fingertips are not detectable in their setting.

## 3 Design and Implementation

In this work, we have developed a multi-finger finding algorithm working with a regular diffuser. A sample installation applying the proposed algorithm is shown in figure 1. The infra-red camera coupled with an infra-red illuminator is settled on one side of the diffuser, observing human gestures taken place on the other side. On user's hands approaching the diffuser, the camera observes reflection left by the hands to do further recognition. The same installation was also found in [1], but they only provide palm-level recognition, so only simple operations were presented. There are several reasons making fingertip recognition behind a diffuser a difficult problem. First, since the diffuser dilutes the reflection with the distance, we can not expect a clear silhouette of hand to be segmented by simply a threshold. Second, it is usually hard to produce a uniform illuminated surface without calibration, so some algorithm tries to well segment hand shape on a region of the surface can be failed on other regions. Third, with a high degree of change in hand gesture, the observed reflection can be in freeform, which leads a simple matching-based approach easily locate fake fingertips.

We roughly classify hand gestures into four cases as shown in figure 2. In the first three cases, fingertips touching the surface are expected identified. In the last case, the hand hovers over the surface, but the reflection on it is still observable, though vague. In this case, no fingertip should be found. The proposed algorithm works well in all the cases.



**Fig. 1.** Four cases of close-up views of hand gestures taken placed on the surface. The first row shows the gesture types. The second row shows the corresponding observations taken by infra-red camera.

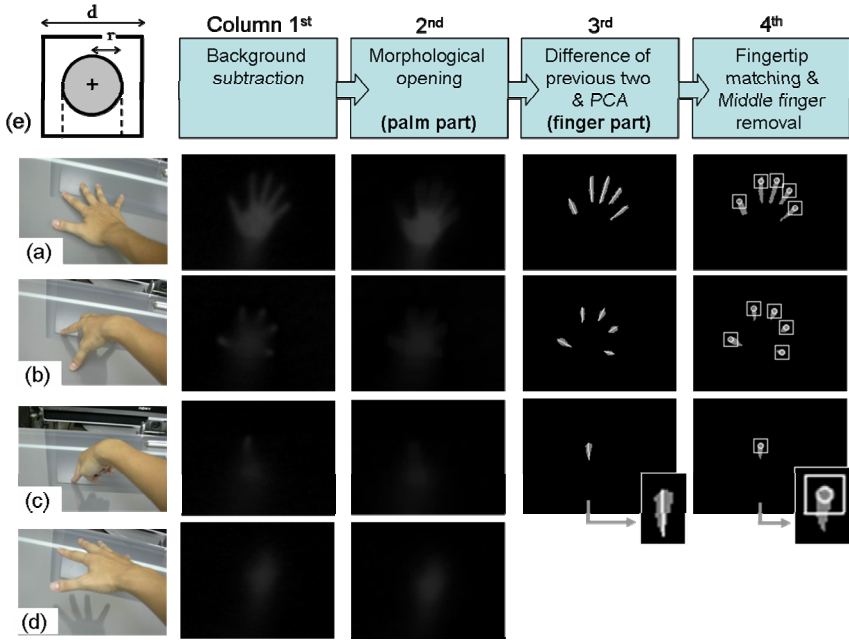
**Finding Fingertips.** Considering a user puts hands on the digital surface (figure 1a-1c), the contact areas of the hands would leave strong reflection, while the other parts dilute with distance away the surface. As a result, the intensity is much solid inside the contact areas, declines rapidly on the boundaries, and finally, shown smooth for the rest. By using the knowledge, our algorithm consists of following steps. First, potential areas are extracted by applying background subtraction. Second, a mathematical morphological opening operation is used to extract watershed from the subtracted images. The watershed is then used to separate finger parts reflection from the rest. For each finger part reflection, we calculate its principal axis and pick points around then two ends of the axis as the candidate positions. At last, fingertip template matching and middle finger removal are used to reject fake candidates. By several rejection steps concatenated, we can effectively locate only a few position candidates for proceeding sophisticated and maybe time-consuming verifications like template matching. The proposed processes well work on a surface suffering non-uniform illumination. For more detail, we describe each step on the following. Refer to finger 2 to find some results produced in each step.

- (a) *Background subtraction*: we first extract potential areas by applying background subtraction.
- (b) *Separating finger and palm parts reflection by a morphological opening*: since the contact areas left by hands usually have strong reflection, we separate finger part reflection from palm part reflection by using a morphological opening with a structure element having its size larger than a normal finger and smaller than a palm. We define a normal fingertip pattern with  $r$  as the radius of circular

fingertip (figure 2e). The size of the structure element for opening is set twice of  $r$ . The 2<sup>nd</sup> Column in figure 2 shows the palm parts of reflection after opening in four cases of gestures. In the implementation, we use a template having a square of  $17 \times 17$  pixels with a circle whose radius  $r$  is 5 pixels.

- (c) *Identifying finger part reflection*: we apply a difference operation between subtraction images and results after opening operation, to extract the finger part reflection. The resulting difference is then dichotomized to the finger regions. Identifying finger region greatly reduces potential area where fingertips might locate. The 3<sup>rd</sup> column in figure 2 shows finger regions are successfully extracted in all cases. Noted that, for the 4<sup>th</sup> case gesture, (figure 2d), there will be no finger region left.
- (d) *Calculating principal axis for each finger part region*: in this step, we further reduce potential area to a principal line by using principal component analysis technique. Positions around one end of the principal line are selected as fingertip candidates and to form a group. Candidates in each group are scored in next step. The survived candidate with best matched score in the group is then selected as fingertip. The direction of principal line is considered as the orientation of the fingertip which will be used in tracking fingertips and allocating fingertips to palms in the next section. The principal lines of finger regions are augmented to potential areas as shown in 3<sup>rd</sup> column in figure 2. This step reduces the search space from a region to a handful points.
- (e) *Rejecting fake fingertips by pattern matching and middle finger removal*: after previous steps, only a few fingertip candidates are passed. In this step, we verify fingertip candidates by using 1) fingertip matching, and 2) middle finger removal, which are two heuristic borrowed from [3] and modified to suit our case. In this step, we verify fingertip candidates using subtracted image (1<sup>st</sup> column in figure 2). In the process of fingertip matching, for each candidate, a template-sized region located at the candidate's position in the subtracted image is copied, and referred as fingertip patch. We then binarize the patch by a threshold set as the average of max and min intensity in the patch. Next, we compute sum of absolute difference between the patch and fingertip template. Candidates with low scores are discarded. In the implementation, we set the score threshold as  $0.8 * \pi * r^2$ . In the process of middle finger removal, if pixels on the edge of fingertip patch coexist in the diagonal direction, then it is not on the fingertip and is removed. Final results are shown in last column in the figure 2.

**Tracking fingertips.** We use Kalman filter technique to track the position and velocity for fingertips. A simple strategy is used to assign detection results (observations) to Kalman trackers. After detection phase finished, each observation creates a searching area to find the nearest tracker according the distance between the observation and the trackers' position predictions. If no tracker is found in its searching area, the observation then creates itself a Kalman tracker. Trackers with on observation fed more than several consequent frames are discarded. With the high detection frames per second, the simple strategy works well.



**Fig. 2.** The images produced during and after processing. Icons labeled (a), (b), (c), and (d) are four cases of gestures. Produced images for each case are arranged in corresponding row. Icon (e) is the fingertip template used in the process. First three columns collect intermediate results during processing. The last column shows the final results.

**Finding palms.** To find palms, we analyze the palm part reflection as in 2<sup>nd</sup> column in figure 2. Since the reflection left by contact fingertips are removed after the grayscale morphological opening applied, strong reflection remains in palm part reflection is mainly left by placed palm or hovering palm. We use the heuristic that dichotomizes the image by a threshold selected as three quarter of max intensity of the reflection. A connected component operation is then applied to the binary image. Mean positions for each component with its size larger than a predefined value is picked as detected palm.

**Allocating fingertips to corresponding palms.** In this section, we describe several steps to associate tracked fingertips to corresponding palms. In general, each palm might have several fingertips associated. Palms with no fingertip associated are users having their hands hovering over the diffuser.

In order to find corresponding palm  $p^*$  for each fingertip  $f$ , following information is required: (1) a pair of fingertip position and direction  $\langle f^{pos}, f^{dir} \rangle$ , (2) a set of palm candidates within proximity of the fingertip  $\{p_1, p_2, \dots, p_s\}$  and their positions  $\{p_1^{pos}, p_2^{pos}, \dots, p_s^{pos}\}$ , and (3) a set of unit vectors gives directions from  $p_i^{pos}$  to  $f^{pos}$  and recorded as  $\{p_1^{dir}, p_2^{dir}, \dots, p_s^{dir}\}$ . The association of fingertips and palms is based on two observations. First, the fingertip would be in proximity of its own palm  $p^*$ . Second, the included angle between  $f^{dir}$  and  $p^{*dir}$  would be small. Figure 5 shows an

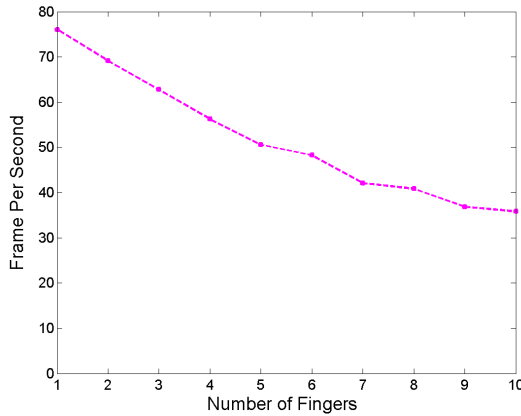
illustration of the idea. Specifically, we define a measure between a fingertip and a palm candidate as follow:

$$m = \arg_i \min \left\{ \left\| f^{pos}, p_i^{pos} \right\| \times \left( 1 - \cos \left( A(f^{dir}, p_i^{dir}) \right) \right) \right\},$$

$$p^* = p_m,$$

where  $A(f^{dir}, p_i^{dir})$  computes the included angle between directions  $f^{dir}$  and  $p_i^{dir}$ .

**Performance Evaluation.** To demonstrate the effectiveness of the algorithm, we include an evaluation on frames per second versus number of fingers simultaneously sliding on the surface. Our experiment was done on a Pentium IV 2.4Ghz machine with 512MB of memory. The video stream from the infra-red camera is processed at 360x240 pixel resolution, covering a full view of the surface (106 cm x 76 cm.). In the figure 3, the resulting curve shows a sub-linear relationship between frames per second and number of fingers simultaneously sliding on the surface. While we have not done careful optimization of the codes, the current implementation achieves more than 70 frames per second in average when single finger shown on the surface and 35 frames per second in ten-finger case.



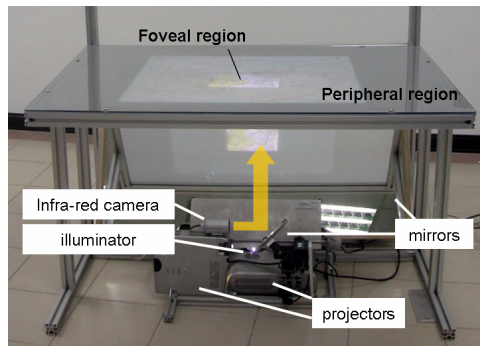
**Fig. 3.** The average frames per second versus the number of fingers moving on the surface

## 4 Application Demonstration

We have developed the proposed finger detection algorithm on a personal tabletop system, named i-m-top. A video retrieval application on the tabletop system is also implemented to help users find interested videos from a large video database. The user is allowed to issue queries, manipulate retrieved results, and feed positive videos back by using barehanded interactions.

#### 4.1 I-M-Top

I-m-top is an interactive rear-projection multi-resolution personal tabletop system (figure 6) which has a diffuser as its tabletop surface. The system includes two projectors, one named foveal projector and the other named peripheral projector, to present a multi-resolution display on the tabletop. The presented display consists of one foveal region on a part of the tabletop in front of the user, and one peripheral region covering the whole tabletop. With the multi-resolution design, the user can have detailed perception in the foveal region while retain overall view of the whole space in the peripheral region. For the detection part, the system has an infra-red camera coupled with an infra-red illuminator installed under the tabletop, observing users' hand gestures taken place on the tabletop. The detection results are then fed to applications of the system.



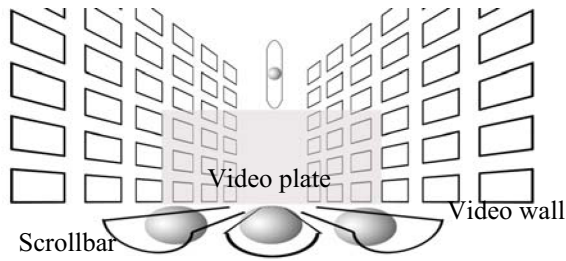
**Fig. 4.** A shot of i-m-top, an interactive rear-projection multi-resolution tabletop system

#### 4.2 A Sample Application – Video Retrieval System

In the following we give a brief introduction to the video retrieval system and more detail description on three main functions for the users to operate with.

When the user issues a query, the application starts by retrieving videos from database and arranges them on two vertical walls in a 3D scene as shown in figure 7. In the center of the space is a transparent plate covered by foveal projection. Videos more relevant to the query are arranged closer to the plate and presented with higher resolution, so the user can easily see and manipulate them. On the contrary, irrelevant videos are arranged over the peripheral region to also give the user a rough view. If interested, the user can drag them onto the plate to obtain a detailed perception of video content. Under the plate are three scrollbars, with which the user can manipulate video walls at will. With the benefits of multi-resolution display, we have larger display region to present more video results to the user at a time. By using fingertips and hovering palms, the user can find interest videos easily. More specifically, we describe three main functions of the application in the following:

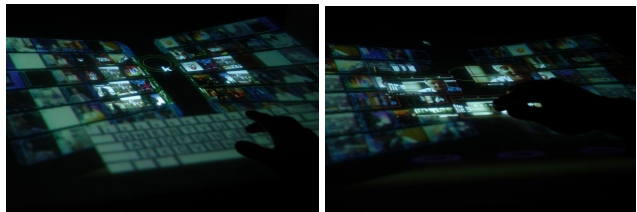
1. *Issue queries*: for this function, we make a virtual keyboard. In the initial, the virtual keyboard is enveloped into a button as [figure 2\(a\)](#). When the button is



**Fig. 5.** The artist's sketch of the video retrieval system. Retrieved videos are presented on the two vertical walls. The video plate in the center gives an area to present detailed video content and to keep positive videos for relevance feedback. The scrollbars on the bottom and top allow users to slide video walls in and out, up and down.

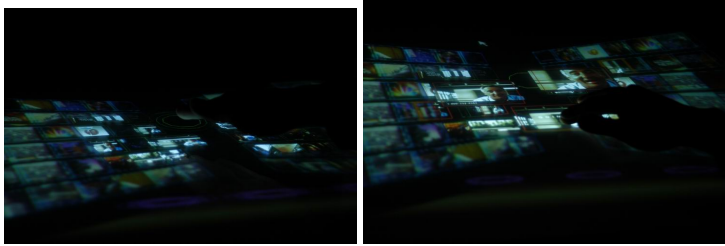
touched by fingertips, the virtual keyboard spreads out. The user then uses fingertips to hit the keys to key in a searching question. Once the “Enter” key pressed, the query is sent out, and the virtual keyboard is enveloped again.

2. *Browse videos:* After query issued, retrieved videos are then arranged on the vertical walls and presented with still key shots. If the user has a palm hovering over a video, the video then turns flat and start playing to give the user a preview of video content. If hovering palm left, the video then stops and backs to its original state. For interest video results, the user can use fingertips to drag them out of the walls as *figure 4(a)*, place on the center plate, and see them clearly. On the bottom, the user can slide fingers on scrollbars placed on two sides and middle of the area, sliding two video walls in and out separately or both. By sliding walls, the use is allowed to have videos in the peripheral region moving into foveal region, or to make video results which are initially invisible flying into the screen (*figure*). Moreover, the scroll bar on the top is for users to move the two walls up and down.
3. *Feedback positive videos:* videos left on the center plate are considered been selected as positive results, the user can press the button on the leftmost of operation area to issue a feedback. New results then replace the videos of the walls.



**Fig. 6.** The left image shows the user is typing on the virtual keyboard. The right image is the user previews a video by having a hovering palm over it.





**Fig. 7.** The user moves fingertips on the scrollbar. The video walls are sliding in, showing more videos onto the surface.

## 5 Conclusion

In this work, we introduced a multi-fingertip finding algorithm that works with a regular diffuser, an infra-red camera coupled with an infra-red illuminator. The algorithm is capable of detecting finger-touch event and palm-hover events when the user interacts with the diffuser. Our experimental results have shown that the performance produces more than 70 frames per second in single-finger case and more than 35 frames per second in ten-finger case. This is important when multi fingers simultaneously operating on the surface to be able to obtain fluent interactions. The installation is quiet simple, cost-effective and flexible to deploy with digital surface achieved by either front or rear projection.

**Acknowledgments.** This work was supported in part by the grants of NSC 95-2422-H-002-020 and NSC 95-2752-E-002-007-PAE.

## References

1. Rekimoto, J., Matsushita, N.: Perceptual Surfaces: Towards a Human and Object Sensitive Interactive Display. In: Proceedings of ACM Workshop on Perceptive User Interfaces (PUI 1997) (1997)
2. Hardenberg, C.V., Berard, F.: Bare-hand human-computer interaction. In: Proceedings of the ACM Workshop on Perceptive User Interfaces (PUI), Orlando, Florida (2001)
3. Koike, H., Sato, Y., Kobayashi, Y.: Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system. In: ACM Trans. Computer.-Human Interact (CHI 2001) vol. 8(4), pp. 307–322 (2001)
4. O'Hagan, R.G., Zelinsky, A., Rougeaux, S.: Visual gesture interfaces for virtual environments. *Interacting with Computers* 14(3), 231–250 (2002)
5. Corso, J., Burschka, D., Hager, G.: The 4D Touchpad: Unencumbered HCI With VICs. *IEEE Workshop on Computer Vision and Pattern Recognition for Human Computer Interaction (CVPR-HCI)* (June 2003)
6. Wilson, A.: TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction. In: International Conference on Multimodal Interfaces (ICMI 2004) (2004)
7. Malik, S., Laszlo, J.: Visual Touchpad: A Two-handed Gestural Input Device. In: Proceedings of the ACM International Conference on Multimodal Interfaces (ICMI (2004) pp. 289–296 (2004)

8. Letessier, J., Bérard, F.: Visual Tracking of Bare Fingers for Interactive Surfaces. In: ACM Symposium on User Interface Software and Technology (UIST 2004), Santa Fe, New Mexico, USA (2004)
9. Han, J.Y.: Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST 2005) (2005)
10. Wilson, A.D.: PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System. In: ACM Symposium on User Interface Software and Technology (UIST 2005), Seattle (October 2005)