Fingertip Interaction With Large Screen Display System Using Virtual Touch Screen

Dejan Chandra Gope[a], Md Nasir Uddin[a], Md Anisur Rahaman[a]

[a]Department of Computer Science and Engineering
Dhaka University of Engineering and Technology (DUET)
Gazipur-1700, Dhaka, Bangladesh.
dejan.gope23@gmail.com

ABSTRACT: Existing large scale display systems generally adopt an indirect approach to user interaction. This is due to the use of standard desktop-oriented devices, such as a mouse on a desk, to control the large wall-sized display. By using an infrared laser pointer and an infrared tracking device, a more direct interaction with the large display can be achieved, thereby reducing the cognitive load of the user and improving their mobility. This paper introduces a novel approach towards direct interaction with large display systems. However, the computer mouse remains the most common interaction tool for such displays. We propose a new approach for fingertip interaction with large display using virtual touchscreen. By taking into account the location of the user and the interaction area available, we can estimate an interaction surface virtual touchscreen between the display and the user. Users can use their pointing finger to interact with the display as if it was brought forward and presented directly in front of them, while preserving viewing angle. An interaction model is presented to describe the interaction with the virtual touchscreen, using the head-hand line method.

Keywords: Large Screen, Face Detection, Hand Gesture Recognition, Virtual touchscreen, Hand Pointing, Pointing Accuracy, Computer Vision, Fingertip Interaction, Human Computer Interaction(HCI), Bare-hand Control, Hand-posture Recognition.

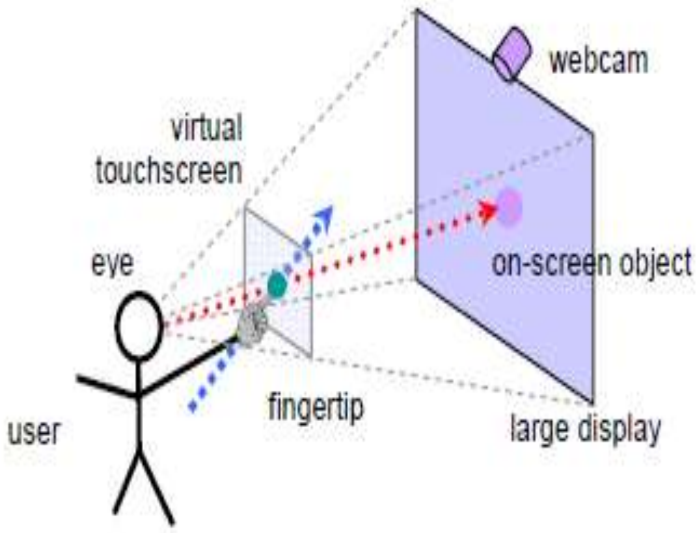
1 INTRODUCTION

Large displays are now more widely used than ever before, yet in most cases, the user interaction still consists of the computer mouse a device that constraint users' mobility. One common approach is to use our own hand as the input method, making pointing as easy as pointing to real world objects. Computer vision can be used to detect and track the user's pointing gesture and is an active area of research. Computer vision based systems have the advantages of being a non-invasive input technique and do not require a physically touchable surface, which is highly suitable for interaction at a distance such as public spaces. To use the mouse, the user needs to place it on a desk or a flat surface. This constrains the user to stay within arm's reach of the table and thus reducing the mobility of the user. To interact with the system, the user moves the mouse horizontally across the table, while the cursor on the display moves vertically. The user also needs to spend a small fraction of time considering how their mouse movements will be mapped onto the large display, although this effect is reduced with practice. Yet another problem is the need to turn around every time to see where the pointer is on the large display. Thus the mouse is not optimal for interacting with large displays A better approach is a system that allows for direct interaction between the user and the object seen on the large display. For example, pointing at the display using fingers or a laser pointer, or to rotate objects by twisting, pushing or turning of the hands. In general, we need a device that allows the user to interact directly with the display, without the need for an intermediary device. Such systems are more natural and easier to use. The computer mouse that is currently being used is a pointing device. It is an intermediary device that facilitates the user, providing a means for humans to interact with the computer. It is a tool for mapping hand movements into an on-screen cursor so that on-screen objects can be manipulated. However, this mapping only provides an indirect interaction. This indirectness comes about because the output space is not the same as the input space. The output space is the display (the monitor in most cases sitting on top of a desk in front of a user) while the input space is the table (the horizontal desk that the mouse lies on). This indirectness causes reduced freedom as well as reduced efficiency. Eye tracking devices follow the movement of the pupils and can in principle be used for cursor control. This technology is beneficial within a confined, controlled space but there is currently no evidence to support its use on a large display. Another promising input system is voice-recognition and although recognition accuracy is improving, these systems are primarily command-based, thus indirect.Touch screens provide an excellent solution to the problem of direct interaction and provides high precision but again, they do not scale well to large displays as manufacturing such large touch display is not feasible. Even if it were possible, the user cannot reach the top ofthe display nor pace across from side to side. The same can be said of specialised whiteboards, such as MimioMouse, Xerox Liveboard and Flatland, which allow the user to press against the wall at specific location on the display with a pen-like device. One device that is attracting an increasing amount of research is the laser pointer. These have the advantages of mobility, direct interaction, and being comparably inexpensive with the notable disadvantages of lag and the instability with the human hand. Many studies into these systems have been carried out. Dwelling is a popular technique for interaction and Olsen investigated the effect of lag with this method which led to a discussion on the use of visible and invisible laser pointers. In this paper, we present a new concept for interacting with large screens remotely using the pointing gesture - by approximating a smaller but invisible screen (virtual touchscreen) that is reachable by the user with an extended arm and fingertip.

2 RELATED WORK

Various vision-based techniques have been used for interaction with large scale displays. For example, the systems presented in and track a laser pointer and use it as an input device which facilitates interactions from a distance. The VisionWand system uses simple computer vision algorithms to track the colored tips of a simple plastic wand to interact with large wall displays both close-up and from a distance. A variety of postures and gestures are recognized in order to perform an array of interactions. A number of other systems use vision to track bare, unmarked hands using one or more cameras, with simple hand gestures for arms-reach interactions. For example, the Bare-Hand system uses hand tracking technology to transform any ordinary display into a touch-sensitive surface. Similarly, the Touchlight system uses two cameras to detect hand gestures over a semi-transparent upright surface for applications such as face-to-face video conferencing or augmented reality. The major advantage of such vision-based techniques is their ability to track multiple fingers uniquely, which allows for more degrees of freedom when compared to standard input devices such as a mouse. However, this advantage of vision-based techniques has not yet been fully leveraged for interactions with wall-sized displays.

The pointing finger has been used in many existing systems. Cipolla & Hollinghurst used a pair of uncalibrated stereo cameras with active contours to track and detect the position and direction of index finger in a 40cm workspace. In the Ishindenshin system a small video camera is placed near the center of a large vertical display directly in front of the user. The user is able to keep eye-contact and use the pointing gesture to interact with another user in a video conference. The fingertip location is detected by the camera and its location in x and y coordinate are determined. Hand tracking is another popular method to support natural interaction. While the above work focuses on vertical screens, Oka, Sato & Koike used hand and fingertip tracking on an augmented desk interface (horizontal). An infrared camera is used to detect areas close to the human body temperature.



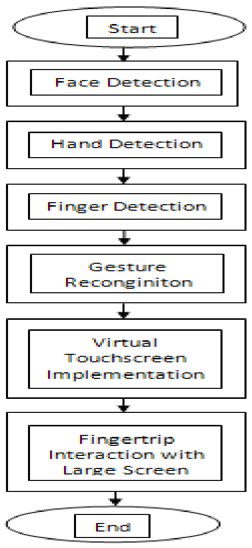

Start
Face Detection
Hand Detection
Finger Detection
Gesture Reconginiton
Virtual Touchscreen Implementation
Fingertrip Interaction with Large Screen
End

Figure-1: Overall Interaction System.

Figure-2: Overall System Flowchart.

3 FACE DETECTION

In recent years, face recognition has attracted much attention and its research has rapidly expanded by not only engineers but also neuroscientists, since it has many potential applications in computer vision communication and automatic access control system. Especially, face detection is an important part of face recognition as the first step of automatic face recognition. However, face detection is not straightforward because it has lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illuminating condition and facial expression.

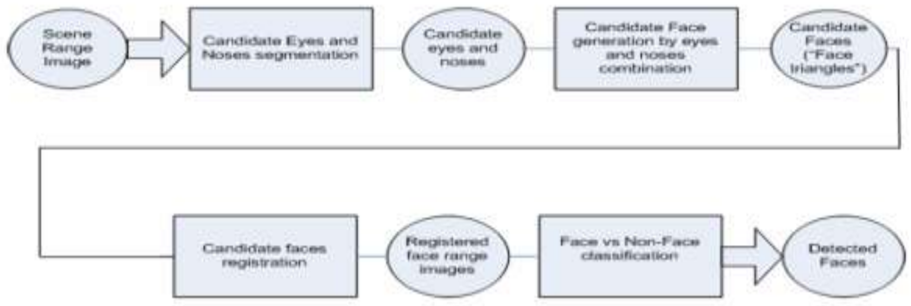

Scene Range Image
Candidate Eyes and Noses segmentation
Candidate eyes and noses
Candidate Face generation by eyes and noses combination
Candidate Faces ("Face triangles")
Candidate faces registration
Registered face range images
Face vs Non-Face classification
Detected Faces

Figure-3: Schematic Diagram of Face Detection Method.

3.1 YCbCr Color Space

YCbCr color space has been defined in response to increasing demands for digital algorithms in handling video information, and has since become a widely used model in a digital video. The Recommendation 601 specifies 8 bit (i.e. 0 to 255) coding of YCbCr, whereby the luminance component Y has an excursion of 219 and an offset of +16. This coding places black at code 16 and white at code 235. In doing so, it reserves the extremes of the range for signal processing foot room and headroom. On the other hand, the chrominance components Cb and Cr have excursions of +112 and offset of +128, producing a range from 16 to 240 inclusively.approach to gestural interfaces, Multimodal User Interfaces, where hand poses and specific gestures are used as commands in a command language. The gestures need not be natural gestures but could be developed for the situation, or based on a standard sign language

3.2 HSI Color Space

Since hue, saturation and intensity are three properties used to describe color, it seems logical that there be a corresponding color model, HSI. When using the HSI color space, you don't need to know what percentage of blue or green is required to produce a color.

3.3 Color Segmentation

Detection of skin color in color images is a very popular and useful technique for face detection. Many techniques have reported for locating skin color regions in the input image. While the input color image is typically in the RGB format, these techniques usually use color components in the color space, such as the HSV or YIQ formats. That is because RGB components are subject to the lighting conditions thus the face detection may fail if the lighting condition changes. Among many color spaces, this project used YCbCr components since it is one of existing Matlab functions thus would save the computation time. In the YCbCr color space, the luminance information is contained in Y component; and, the chrominance information is in Cb and Cr. Therefore, the luminance information can be easily de-embedded. The RGB components were converted to the YCbCr components using the following formula.

$Y = 0.299R + 0.587G + 0.114B$

$Cb = -0.169R -0.332G + 0.500B$

$Cr = 0.500R -0.419G - 0.081B$

3.4 Image Segmentation

The next step is to separate the image blobs in the color filtered binary image into individual regions. The process consists of three steps. The first step is to fill up black isolated holes and to remove white isolated regions which are smaller than the minimum face area in training images. The threshold (170 pixels) is set conservatively. Finally, the previous images are integrated into one binary image and relatively small black and white areas are removed. The difference between this process and the initial small area elimination is that the edges connected to black areas remain even after filtering. And those edges play important roles as boundaries between face areas after erosion.

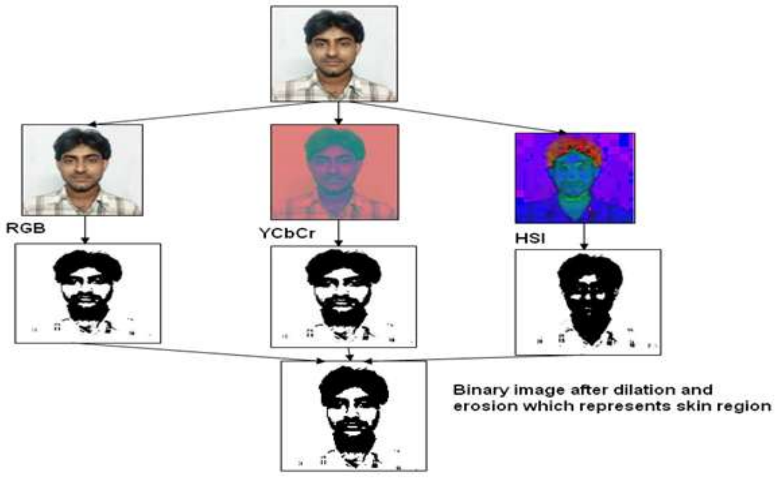

RGB
YCbCr
HSI
Binary image after dilation and erosion which represents skin region

Figure-4: Skin color detection from proposed algorithm.

3.5 Face Triangle Calculation

Face detectionis concerned with finding whether or not there are any faces in a given image (usually in gray scale) and, if present, return the image location and content of each face.This is the first step of fingertip interaction with large display using virtual touchscreen that analyzes the information contained in faces (e.g., identity, gender, expression, age, race and pose). While earlier work dealt mainly with upright frontal faces, several systems have been developed that are able to detect faces fairly accurately with in-plane or out-of-plane rotations in real time. Although a face detection module is typically designed to deal with single images, its performance can be further improved if video stream is available.

Figure-5: Three points (i, j, and k) satisfy the matching rules, which will form an isosceles triangle.

4 HAND AND FINGER DETECTION

Human hand and finger detection is an important role in human-computer interface. They usually function through keyboard, mouse, digitizer, touch panel, and so on. There are many related works of fingertrip interaction with large screen dispaly which utilize free hand/finger tracking: interface in VR environment , television control and finger pointer. In this paper, we describe barehanded interaction between human and large screen display for Hand and Finger Detection. Barehanded means that no device and no wires are attached to the user, who controls the computer directly with the movements of his/her hand. We identify three essential services for fingertrip interaction with large screen dispaly: detection, identification and tracking. We will briefly present the three services and describe how they are used by our envisaged applications.

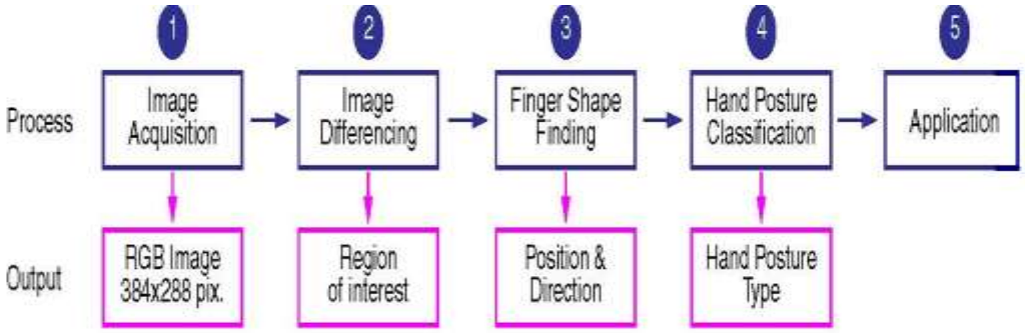


Figure-6: The Hand Posture Recongnition and Finger Findind Process.

4.1 Hand Tracking

This section describes the implementation details of the hand tracking system, which is primarily based on the single hand tracker. The system can extract the 3D position and 2D orientation of the index finger for each hand, and when present the pose of the thumb as well. In interactive applications a single pointing gesture could then be used for selection operations while both the thumb and index finger could be used together for pinching gestures in order to grasp and manipulate virtual objects.

4.2 Background Substraction

The first phase of the tracking system involves separating potential hand pixels from non-hand pixels. Before segmentation occurs, we first convolve all captured images with a 5x5 Gaussian filter and then scale this filtered image by one half in each dimension in order to reduce noisy pixel data. All subsequent image processing operations are then performed on this scaled and filtered image. Since the stereo cameras are mounted above a non-moving workspace, a simple background subtraction scheme is used to segment any potential foreground hand information from the non-changing background scene. Figure 1 shows the result of background subtraction on an image containing a hand.

4.3 Skin Segmentation

Although the background subtraction scheme described about works fairly well in segmenting foreground data from the non-changing background, it will still allow objects such as shirt sleeves, coffee mugs, or other desktop items that are placed into the workspace to be detected as potential hands. In order to deal with such situations and add some more flexibility to the system, a skin pixel detector has been implemented to further filter the foreground data. As a pre-processing step, for each camera a small number of snapshots are taken of various hands with a range of different skin-tones and poses. Then using an image editing program each of the captured images is manually segmented into a binary mask where white pixels represent skin areas and black pixels represent non-skin areas. This set of captured images and associated skin masks is then used as the training set for a histogram-based skin classifier. Figure 2 shows the result of skin detection using the background subtracted image.

4.4 Region Extraction

Now that the skin regions have been detected, we must determine which regions correspond to the left and right hands. It is possible that small noisy regions will still be present after background subtraction and skin segmentation, but we assume that the regions corresponding to the hands will be the largest. Thus we first extract the contours of all the detected skin regions using binary image processing operations and connected component analysis.

4.5 Feature Extraction

In order to find the fingertips for the thumb and index finger for each hand, we attempt to find pixels that represent peaks along the contour perimeters. the points as either peaks or valleys, we convert the vectors into 3D vectors lying in the xy-plane and then compute the cross product. If the sign of the z component of the cross product is positive then we label the point as a peak, while a negative cross product results in a valley label. Finally, non-maximal suppression is then used to find the strongest peaks and valleys along the perimeter, since we can expect that a sequential set of peaks and valleys will be detected in the neighbourhood of the strongest locations. Figure 4 shows the result of peak and valley detection on the image in Figure 3. Note that not all valleys were detected, largely as a result of the morphological closing operation that was performed during skin segmentation.

5 HAND GESTURE RECOGNITION

Hand gesture recognition system to recognize real time gesture in unconstrained environments. Our gesture alphabet consists in four hand gestures and four hand directions in order to fulfil the application's requirements. The hand gestures correspond to a fully opened hand (with separated fingers), an opened hand with fingers together, a fist and the last gesture appears when the hand is not visible, in part or completely, in the camera's field of view. These gestures are defined as Start, Move, Stop and the No-Hand gesture respectively. Also, when the user is in the Move gesture, he can carry out a Left, Right, Front and Back movements. For the Left and Right movements, the user will rotate his wrist to the left or right. For the Front and Back movements, the hand will get closer to or further of the camera. Finally, the valid hand gesture transitions that the user can carry out are defined.

6 FINGER DETECTION

The tracking program allowed the user to define the workspace, move the "mouse" around within the workspace and have the corresponding movement echoed within the application program, and to select screen objects by "clicking" on them. The user's workspace needed to be defined so that accurate relative positions could be mapped to the screen. The user set up the workspace at the beginning of the session by moving the forefinger to the upper-left and lower-right hand corners of the workspace, and performing the clicking action (see Figure 4). This provided the reference coordinates for the workspace area. After setting up the workspace, the user could move the mouse cursor around the application program by moving their finger within the defined workspace, and select objects by performing the click action. Double-clicking was achieved by repeating the click action within a short time. The tracking system was set up as a server on VxWorks and passed event calls to application programs as they occurred. This is similar to the way a mouse works. Changes in the position of the finger were sent by the server using Xevents to the application programs. When "clicks" and "double-clicks" were detected, the change in state was also passed to the application. The Finger Tracker interface was trialed with programs such as xv and xfig. This showed that the finger could be tracked in real-time with cursor movement on the screen providing feedback to the user. This was sufficient to test the robustness of the tracking system.

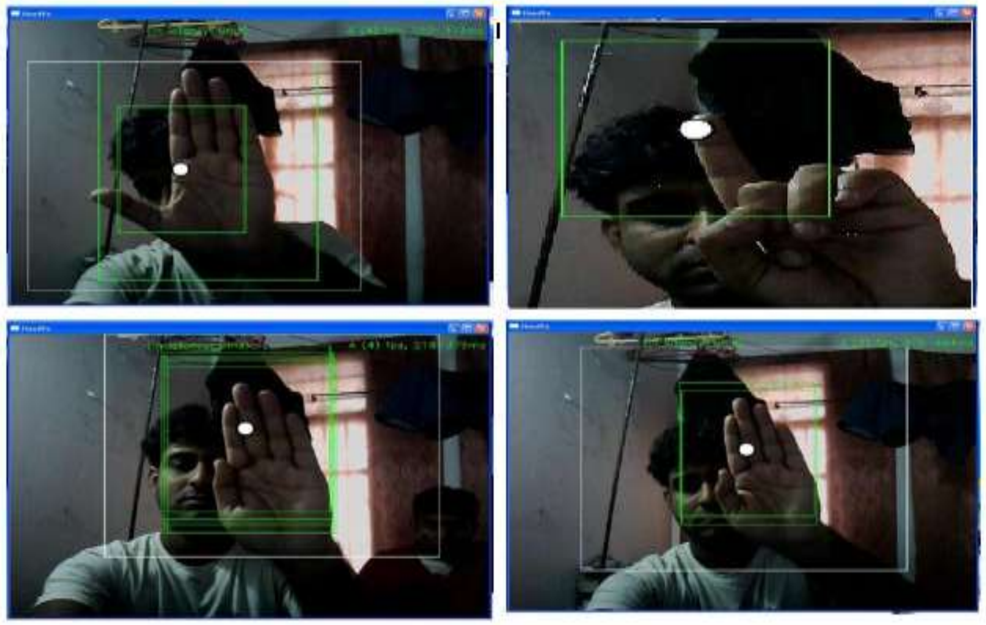


Figure-7: Finger Tracking

7 MOVING MOUSE CURSOR

We used the index finger as a cursor controller to control mouse cursor. We used two different approaches for moving the mouse cursor. The first method is mapping cursor control. It means that the index finger on a camera screen can position maps to a desktop screen position. In other words, the mouse cursor is placed on the desktop window along with the index finger tips position displayed on the camera screen position. This method has a problem. If the resolution of the desktop window has a higher resolution than the camera resolution, then the cursor position cannot be accurate because when the camera resolution converts to the desktop window resolution we lose intermediate value. We expect the ratio of jumping pixel is up to 4 pixels. The second method is weighted speed cursor control. We get a difference of the finger of the current image and the previous image and compute the distance between the two. Next, we move the mouse cursor if the gap between the two finger images (current and previous frame) is far then the mouse cursor moves fast or, if the gap is close then the cursor moves slow. This algorithm also has a problem. Since we are working with real-time image processing, the client machine should be fast. The image input speed is 15 frames per second and we need image processing time on CPU. It means that some machines which cannot achieve image processing 15 images per sec do not work smoothly because computing the image center and the hand shape takes time. Thus, this algorithm does not work properly. In this paper, we used the first method which uses absolute position of finger tips because it is more accurate than the second method.

7.1 Left Clicking and Double-Clicking

To call system event for left clicking, at least two convex hull vertexes have to be off the palm area which was computed in the previous part. In addition, the x position of one of two vertexes should be lower than the other to restrict detection of other finger tips. When the degree of the index finger and thumb is 70 to 90 degree then we can recognize that the gesture is left clicking. Actually, if the thumb is placed outside the circle of the hand region, then the gesture is left clicking. The double-clicking occurs when the thumb moves 0 to 90 degree and back two times fast.

7.2 Right Clicking

We simply implemented this part using previous gestures. If we make the hand pose left clicking for 3 seconds, then the system calls the right clicking event.

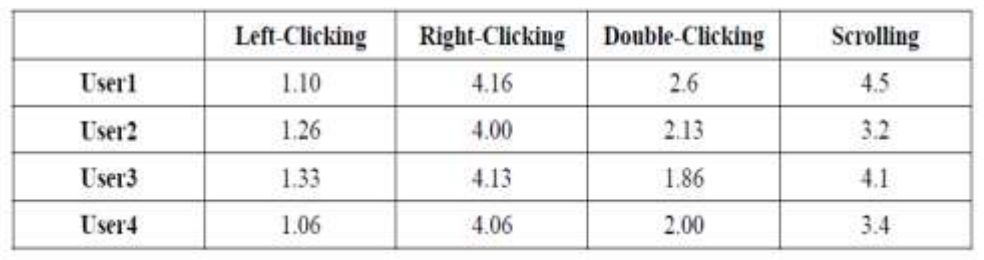
7.3 Scrolling

Scrolling is a very useful task with hand gestures. We also implemented this feature in our system. As figure 5 shows, we assigned a 10% margin as a scroll area on the right side of the desktop screen region and divided by half. If the index finger placed the upper area of the scroll area with clicking pose then the system called the scroll up function or else the system called the scroll down function.

8 EXPERIMENTS AND RESULTS

We tested all mouse tasks such that left click, right click, double-click, dragging, and scrolling on windows. We could not compare with the mouse device because this hand gesture system

always shows lower performance than real mouse device. Instead of comparing with a real mouse, we allowed to use this system to four testers to know how it can be adapted easily.
Our method for evaluating performance is that we checked the time that a task, such as clicking, double clicking and scrolling, is done. We designed four experiments to get performance. In the first experiment we placed an icon on the center of desktop window and put the cursor in the top-left corner. We then measured the time in how long it takes to select the icon. In the second experiment the position of the icons is same and we only measured the time to show the drop down menu on the icon. In the third experiment the position of the icons is same with the second experiment and we measured the time to open the icon. In the final experiment we opened a web site (http://news.yahoo.com) on the center of the desktop window and measured the time until the scroll bar moves from top to bottom. The results are shown below: (time = sec)



	Left-Clicking	Right-Clicking	Double-Clicking	Scrolling
User1	1.10	4.16	2.6	4.5
User2	1.26	4.00	2.13	3.2
User3	1.33	4.13	1.86	4.1
User4	1.06	4.06	2.00	3.4

Table-1: The average time of all experiment results. Four users tested clicking, double-clicking, and scrolling. Measuring the time until finishing a given task, we estimated the system performance.

Every tester was new in this system. The clicking task showed similar times with all testers. However, the scrolling result showed the time as unstable. The reason is that the focus of the camera lens works automatically so when the index finger went to the border on the screen then part of the hand colors becomes dim. Eventually, hand segmentation failed because colors could not be on the color range which is in the segmentation part. Thus, we could not get a clear shape of the hand from the video streaming.

9 LIMITATIONS AND CONSIDERATIONS

1.Latency problem. Which define the time gap between an action of the user and the system response. The minimum acceptable latency for real-time applications is in the area of 50ms, resulting in a required minimum processing frequency of 20Hz.
1.Resolution problem. For gesture interfaces, the output resolution does not affect the quality of the service, but detection and identification processes require a minimum input resolution.
2.Stability problem. There are several possible sources of instability, such as changing light conditions, motion of distracting objects and electrical noise.
3.Finger movement problem. Finger move slowly or fastly.
4.The arm's length is assumed to be constant. To allow for different users to the system, this length must be measured and adjusted manually.
5.The distance from the hand to the screen remains an unsolved problem.
6. Assume that all users have similar face size, or Introduce a calibration phase at the beginning to measure (manually or automatically) the real width of the user's face.

10 CONCLUSION AND FUTURE WORK

We have presented a new approach for fingertrip interaction with large screen display using virtual touchscreen. The concept of virtual touchscreen was presented and the interaction model associated with it discussed. A work- in-progress prototype using a desktop monitor was used to demonstrate the feasibility of our idea. It is anticipated that this work could be applied similarly to interfaces such as televisions and other devices in the home and workplace. It is also envisioned that the virtual touchscreen can, not only be used for large displays but, also be used for desktop monitor, providing capability similar to a normal touchscreen at close range. We developed a system to control the mouse cursor using a real-time camera. We implemented all mouse tasks such as left and right clicking, double clicking, and scrolling. This system is based on computer vision algorithms and can do all mouse tasks. However, it is difficult to get stable results because of the variety of lighting and skin colors of human races. Most vision algorithms have illumination issues. From the results, we can expect that if the vision algorithms can work in all environments then our system will work more efficiently. This system could be useful in presentations and to reduce work space. In the future, we plan to add more features such as enlarging and shrinking windows, closing window, etc. by using the palm and multiple fingers. This work investigated a number of techniques for interacting with large displays from afar using a visionbased hand and finger tracking system. By allowing users to sit comfortably at a table in front of a large display, traditional selection and navigation techniques become inefficient and other more appropriate methods must be developed. We presented a set of such approaches that leverage people's natural abilities to manipulate real-world items with their hands asymmetrically. Our current design satisfies our original design principles of: (1) leveraging two hands and multiple fingers for both natural and high degree of freedom input, (2) allowing fast targeting to any part of the display, (3) maximizing comfort for from a far interactions, and (4) supporting multiple users. In the future, we would like to investigate how to further integrate multiple users onto a large display using our system. In particular, with the high degree of freedom input provided by two hands, it would be interesting to investigate what sort of collaborative tasks could be performed by two or more users working together. Another fruitful direction for research might be to investigate how vision algorithms could be further leveraged for tasks other than just detecting hands. We could very easily place other objects onto the touchpad surface such as documents or other tangible objects, and then project them onto the large display. This opens up the possibility of using real tools to perform virtual tasks in more natural ways.

REFERENCES

Cheng, K. & Takatsuka, M. (2006) Estimating Virtual Touchscreen for Fingertip Interaction with Large Displays. In Proc. OZCHI 2006 (Sydney), ACM Press., 397- 400.

Chen, M. C., Anderson, J. R., and Sohn, M. H., "What can a mouse cursor tell us more? Correlation of eye/mouse movements on web browsing," in Proceedings of CHI'01 conference on Human factors in computing systems, Seattle, ACM, 2001.

Cheng, K. and Pulo, K., "Direct Interaction with large-scale display systems using infrared laser tracking devices," in Proceedings of Proceedings of the Australian Symposium on Information Visualisation (invis '03), Adelaide, ACS Inc., 2003, pp. 67-74.

Cheng, K. and Takatsuka, M., "Real-time Monocular Tracking of View Frustum for Large Screen Human-Computer Interaction," in Proceedings of Proceedings of the 28th Australasian Computer Science Conference, Newcastle, Australia, 2005.

Cheng, K. and Takatsuka, M., "Estimating Virtual Touchscreen for Fingertip Interaction with Large Displays," in Proceedings of Proc. OZCHI 2006, Sydney, ACM Press., 2006, pp. 397-400.

Cheng, K. and Takatsuka, M., "Hand Pointing Accuracy for Vision-Based Interactive Systems," in Proceedings of INTERACT 2009, 12th IFIP TC13 Conference in
Human-Computer Interaction, Uppsala, Sweden, Springer Verlag (to appear), 2009.

Cheng, K. and Takatsuka, M., "Initial Evaluation of a Bare-Hand Interaction Technique for Large Displays using a Webcam," in Proceedings of EICS'09, ACM SIGCHI

Symposium on Engineering Interactive Computing Systems, Pittsburgh, Pennsylvania, ACM Press. (to appear), 2009.

Cheng, K. and Takatsuka, M., "Interaction Paradigms for Bare-Hand Interaction with Large Displays," in Human-Computer Interaction, A. Lazinica, Ed., Vienna, IN-TECH (accepted for publication), ISBN 978-953-7619-26-8, 2009.

Cheng, K., Vronay, D., and Yu, F., "Interaction Design for the Media PC," in Proceedings of Companion proc. UIST'04, Santa Fe, New Mexico, ACM Press., 2004, pp. 5-6.

Cipolla, R. and Hollinghurst, N. J., "A Human-Robot Interface using Pointing with Uncalibrated Stereo Vision," in Computer Vision for Human-Machine Interaction, R. Cipolla and A. Pentland, Eds., Cambridge University Press, 1998, pp. 97-110.

Colombo, C., Bimbo, A. D., and Valli, A., "Visual Capture and Understanding of Hand Pointing Actions in a 3-D Environment," IEEE Transactions on Systems, Man, and Cybernetics, 33(4), University of Florence, Italy, pp. 677-686, August 2003.

Dix, A., Finlay, J., Abowd, G. D., and Beale, R., Human-Computer Interaction, 3rd ed.Pearson Education Limited, 2004.

Baudel, T. and Beaudouin-Lafon, M. 1993. Charade: remote control of objects using free-hand gestures. Commun. ACM 36, 7 (Jul. 1993), 28-35.

Grossman, T., Wigdor, D., and Balakrishnan, R. 2005. Multi-finger gestural interaction with 3D volumetric displays. ACM Trans. Graph. 24, 3 (Jul. 2005), 931-931.

Malik, S. and Laszlo, J. 2004. Visual touchpad: a twohanded gestural input device. In Proceedings of the 6th international Conference on Multimodal interfaces (State College, PA, USA, October 13 - 15, 2004). ICMI '04. ACM Press, New York, NY, 289-296.

J. Serra, *Image Analysis and Mathematical Morphology*, New York: Academic Press, 1983.

K. Homma and E.-I. Takenaka, "*An image processing method for feature extraction of space-occupying lesions*," journal of Nuclear Medicine 26 (1985): 1472-1477.

K. Homma and E.-I. Takenaka, "An image processing method for feature extraction of space-occupying lesions," *journal of Nuclear Medicine* 26 (1985): 1472-1477.

A Method for Controlling Mouse Movement using a Real-Time Camera

A Robust Skin Color Based Face Detection Algorithm

Cahi, D. and Ngan, K. N., "Face Segmentation Using Skin-Color Map in Videophone Applications," *IEEE Transaction on Circuit and Systems for Video Technology*, Vol. 9, pp. 551-564(1999).

Kjeldsen, R. and Kender., J., "Finding Skin in Color Images," *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 312-317 (1996).

Yang, J., Lu, W. and Waibel A., "Skin Color Modeling and Adaptation," (CMUCS-97-146,) CS Department, CMU, PA, U.S.A. (1997).

Yang, M. H. and Ahuja., N., "Detecting Human Faces in Color Images", *Proceedings of IEEE International Conference on Image Processing*, Vol. 1, pp. 127-130 (1998).

Dejan Chandra Gope is a student in the Department of Computer Science and Engineering at the University of Dhaka University of Engineering and Technology, Bangladesh. He received Bachelor of Science in Computer Science and Engineering (B.Sc) from the University of Dhaka University of Engineering and Technology in 2012. He is now studying Master of Science in Computer Science and Engineering (M.Sc) at the same University. His research interest lie in studying the role of computer vision and advanced human-computer interaction, Artificial Intelligence and Pattern Recognition.