1-1-2017

Finger-stylus for non touch-enable systems

Ankit Chaudhary
Independent consultant, chaudharya@umsl.edu

January 1, 2017

Finger-stylus for non touch-enable systems

Ankit Chaudhary, *Independent Researcher, Iowa City, IA 52242, United States*

Journal of King Saud University – Engineering Sciences (2017) **29**, 35–39

King Saud University

Journal of King Saud University – Engineering Sciences

www.ksu.edu.sa
www.sciencedirect.com

TECHNICAL NOTE

Finger-stylus for non touch-enable systems

Ankit Chaudhary

Independent Researcher, Iowa City, IA 52242, United States

Received 30 April 2013; accepted 18 February 2014
Available online 11 July 2014

KEYWORDS

Finger-stylus;
Tablet apps;
Paint;
Screen drawing;
Natural computing;
Fingertip detection

Abstract Since computer was invented, people are using many devices to interact with computer. Initially there were keyboard, mouse etc. but with advancement of technology, new ways are being discovered that are quite common and natural to the humans like stylus for touch-enabled systems. In the current age of technology, the user is expected to touch the machine interface to give input. Hand gesture is used in such a way to interact with machines where natural bare hand is used to communicate without touching machine interface. It gives a feeling to the user that he is interacting in a natural way with some human, not with traditional machines. This paper presents a technique where the user need not touch the machine interface to draw on the screen. Here hand finger draws shapes on monitor like stylus, without touching the monitor. This method can be used in many applications including games. The finger is used as an input device that acts like a paint-brush or finger-stylus and is used to make shapes in front of the camera. Fingertip extraction and motion tracking were done in Matlab with real time constraints. This work is an early attempt to replace stylus with the natural finger without touching the screen.

1. Introduction

Communication using hand gestures exists in all civilizations since old times. There is a specific way to present hand to show a particular message. A gesture is a form of non-verbal communication in which any message is conveyed with the help of visible body actions. Hand gesture is potentially a very natural and useful modality for human–machine interaction. Hand gesture recognition (HGR) now has become an adoptable and reliable way to communicate with machines (Chaudhary et al., 2011a). People are using it to control robots (Chaudhary et al., 2011b), to learn/interpret sign languages (Liang et al., 1998; Starner and Pentland, 1995; Bragatto et al., 2006; Cooper, 2012), in health care (Chaudhary and Raheja, 2013) and many other fields. The use of hand gesture, as an interface between human and machines has always been a very attractive alternative to the conventional interface devices. HGR has been applied to many applications using different techniques since last three decades. Till date, mostly it is sensors or touch based recognition on devices which is used for interaction. The sensor-glove based method hinders the ease and naturalness with which humans interact with the computer. This has led to an increased interest in the visual approach.

E-mail address: dr.ankit@ieee.org

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksues.2014.02.002

These days tablets and touch enabled mobiles are in use and mostly users go in for these devices like Apple iPad, Samsung Galaxy etc. They have few applications (apps) which are controlled or run by touching the screen in a normal way or in a specific way for example minimization of apps, entering data, and paint. It may be a single finger gesture or a multi finger gesture. Here to start with, we are focusing on one application which is paint. It is like 'Paper' or 'Fingerpaint Magic' available for iPad. They work the same as 'Paint' works on Microsoft Windows. There may be many applications like these on different platforms. A detailed analysis of paint with different computer vision algorithms has been done by Booch (2001). Forsline and Pedersen (2004) came up with a stylus to use a pen kind e-stick to draw or write on a computer screen which changes the world of interaction with computers. *Sensu* is a brush with stylus which works on these devices and gives a feeling of painting on a canvas.[1] After this, sensors detecting human body touch on the screen changed the technology of interaction. It seems very natural to draw or point on machines using sensors for example interaction with iPad.

In touch enabled machines, apps developed on different platforms provide flexibility to draw anything on computer canvas (which is screen) using hand fingers. Even few applications use finger pressure to decide the thickness of the paint brush. Among different parts of the body, hand is the easiest to use and shows the expression of human feelings. Also it is very robust in its operations because of its design and can move in any direction. A good comparison between stylus and hand touch devices is given in.[2] Here we are presenting a method to perform the same action by not touching the screen. The paper discusses the implementation of paint drawing on computer screen in real time using a vision based method, where touch-enabled device is not required. Here hand finger works as the stylus, say *finger-stylus*. This can be used on different tablets and replace many existing apps as discussed above because of its easy usage and mostly all tablets have a camera.

2. Background

The robust tracking of hand has been an active area of research in the applications where finger movements or hand geometry detection is needed. The existing methods are generally divided into two categories: Vision based approach (Sudderth et al., 2004) and Glove based approach (Wang and Popovíc, 2009). Both of them have their own pros and cons. Rehg and Kanade (1993) presented a method to detect articulated hand motion. He proposed 27 degree of freedom of hand in gray images but his method was not effective with complex backgrounds. Sato et al. (2000) used infrared cameras for skin segmentation on a table top and template matching for interpreting gesture commands. Chaudhary et al. (2011b) have also developed a real time finger motion detection framework which can be applied to many applications.

Garry (Berry, 1998) used virtual environments with gestures to control it in a natural way. Zeller et al. (1997) also used hand gestures in virtual environments. Starner and Pentland (1995) developed a system with a single color camera to track the American sign language in real time. Bragatto et al. (2006) translated the Brazilian sign language from video by tracking hand gestures. He used perceptron ANN for color segmentation and then classification separately. Cooper (2012) presents a method to control a complex set of sign language with *viseme* representation to increase the lexicon size. Ju et al. (1997) used hand gestures for analyzing and annotating video sequences of tech talks. In his work, gestures like pointing and writing were detected and recognized. Even systems which can detect the hand pointing spot have been developed (Raheja et al., 2014).

Araga et al. (2012) presented a real time gesture recognition system from video where they used few gesture images as gesture states. Wachs et al. (2008) developed a gesture based tool for sterile browsing of radiology images. A robust object detection method in indoor and outdoor field is described by Wang et al. (2011). The hand gesture could replace stylus on touch screens or touch screen sensors which are currently used in many places. A similar gesture based patient care system is described in Chaudhary and Raheja (2013). Coral Inc. has developed fingertip paint brush for touch enabled systems.[3] Hettiarchchi et al. (2013) presented *FingerDraw* where children can paint on an interactive screen with 'worm finger' device.

Figure 1 Finger captured by camera.

3. System description

The system uses a webcam either in-built in system or plugged. It tracks the hand movements made by the user by detecting the finger tips. These tips are displayed on the screen and finally the system shows the whole movement made by the user on the screen by connecting them. This movement further can be used to interact with computers and mobile devices without any physical intervention. For the simplicity of the system and to keep it near to the human behavior, we are considering only one finger for the tracking and display as one brush on a canvas. Later multiple criteria can be added to make the system for more applications.

Generally humans use the index finger to point or to gesture in their daily life. A 'session' is the time when the finger comes in front of the camera view and goes out of its capturing frame.

1 https://www.sensubrush.com.
2 http://purple-owl.com/drawin.
3 http://www.corel.com/corel/product/index.jsp?pid = prod3720128&cid = catalog20038&segid = 534&storeKey = us &languageCode = en.

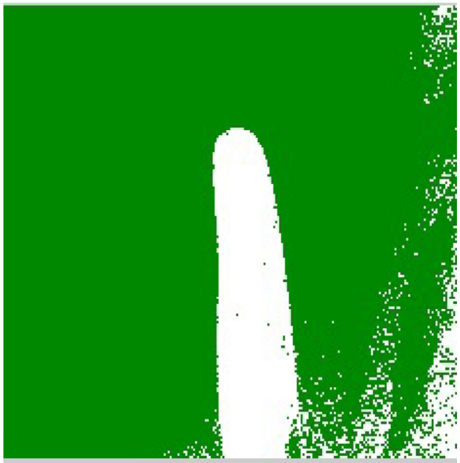

Figure 2 Pre-processing after skin filter.

Figure 3 Fingertip detection results.

If the user again brings a finger in the capturing zone, the system will again start its display. The user must show only one finger (any finger) as shown in Fig. 1.

The system is implemented in Matlab considering real time constraints. The user is free to move its hand in whatever direction and whichever shape he wants to draw in the air, as virtual drawing. Hand image is captured in RGB color space and YC_bC_r color space based skin filter was applied to minimize color variations. Also segmentation is working robustly irrespective of light intensity changes. The pre-processing results are shown in Fig. 2.

Different fingertip detection methods were tested to check real time response including curvature based, and edge based and finally we came up with a faster method which is described in Raheja et al. (2011a). The results are shown in Fig. 3. This method is a direction invariant and fastens pre-processing by cropping the image by a factor more than 2 depending on the skin pixels in the captured image.

4. Implementation

As the system usage, the user should show one finger to the system. If he shows more, the first finger from the left would be chosen. Also it is advisable that the finger should point upward although the developed method is able to detect fingertip in any direction. The skin filter was applied on a captured image frame and one finger chosen was considered. Fingertip detection was done by applying an increasing constant to all pixels to the image from the wrist to fingertip. The constant ranges from 0 to 255 starting from the wrist. Where pixels have values as 255, it would be detected as fingertips.

Mathematically the process can be defined as follows:

$$\text{Finger}_{\text{edge}}(x,y) = \begin{cases} 1 & \text{if modified image}(x,y) = 255 \\ 0 & \text{otherwise} \end{cases}$$

As we are taking only fingertip pixels which are on the boundary of finger, other pixels or noise is very less to occur and we would get a sharp fingertip. Also a finger template matching was done to get the first finger from the captured image, this also helps in noise reduction. To increase the width of fingertips -5 to $+5$ pixels are considered as fingertip and marked as red color. This would be the thickness of line in drawing. It needs to be changed, if someone needs a different thickness in the drawing.

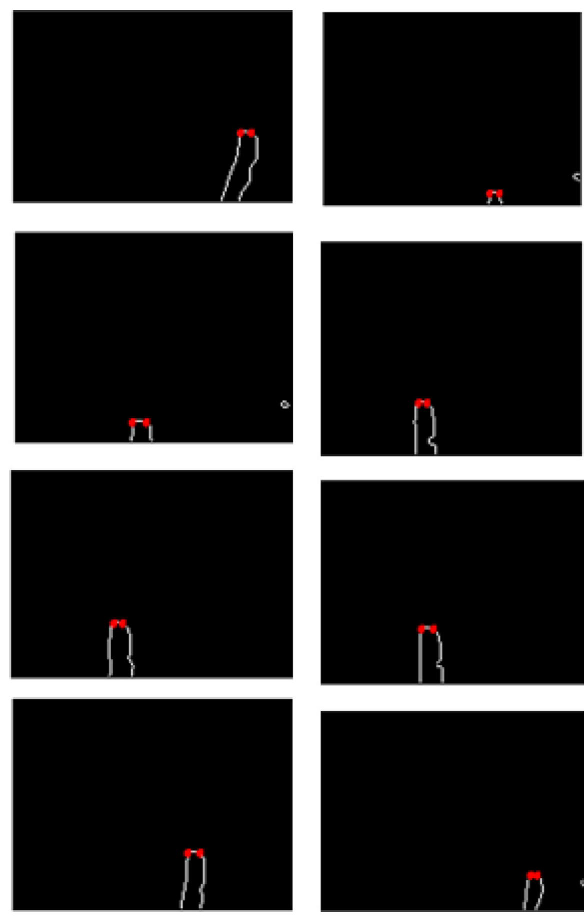

Figure 4 Highlighting fingertips.

The implementation supports real-time responses and gives a feeling of a finger paint brush. The detected fingertips in real time on computer screen are shown in Fig. 4.

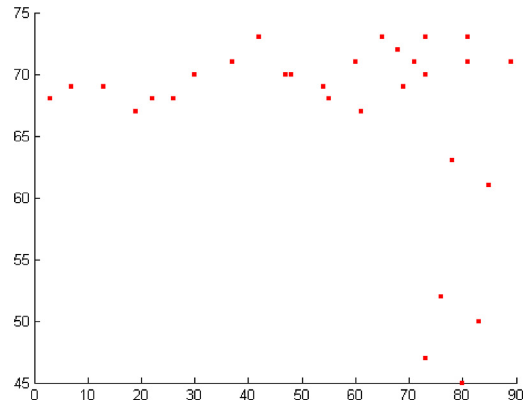

Figure 5 Displaying fingertips tracked on screen in one session.

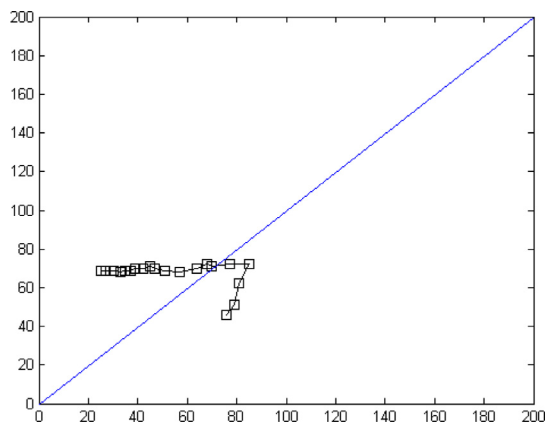

Figure 6 Gesture made by the user on the screen.

All fingertips' pixels tracked in the process, were stored as of that session and then shown on the screen. A movement of finger in one session is shown in Fig. 5. All pixels on the screen were connected by lines, in the same order, in which the finger was moved. The connected lines are shown in Fig. 6. These lines will form the image on the screen to show which user was drawing in air.

In Figs. 5 and 6 the screen positions are shown. The X and Y axes are mapped to the resolution of the monitor screen.

The current methods for interaction are different where they use gloves or sensors. So a direct comparison of our work is not possible, although the method used in this work can be compared with existing implemented methods. Bragatto et al. (2006) proposed a method for sign language, which works well with a recognition rate of 99.2%. Araga et al. (2012) shows the accuracy of 99.0% for 5 different hand postures whereas it obtained an accuracy of 94.3% for 9 gestures. Touch enabled systems where stylus or human fingers are used, have 100% accuracy of gesture recognition (Hettiarchchi et al., 2013).[1,3] A comparison of available methods is shown in Table 1.

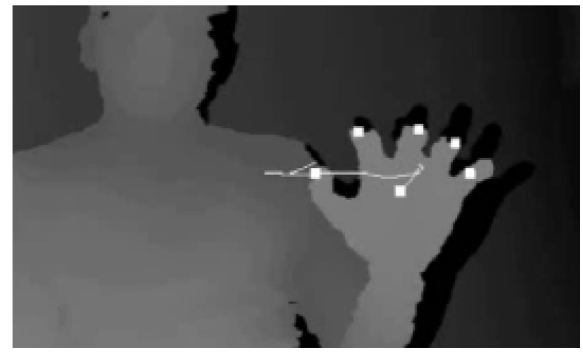

Figure 7 Results with MS Kinect.

With the presented method, we are getting around 100% accuracy with simple backgrounds and 96% correct results on complex backgrounds. To make finger-stylus segmentation to be a light intensity invariant, we tried to implement several methods which are discussed in Raheja et al. (2011b), Chaudhary and Gupta (2012). The system works fine with varying light intensity, although a minimum threshold of light intensity is needed. The system takes 116 ms to show the first print of finger on screen and after that it continuously draws as the system is real time. The system was implemented with Matlab on Windows PC and webcam was capturing images as 12 fps.

Also this system was implemented with MS Kinect where the center of palm was used to draw on the screen as shown in Fig. 7. Kinect based system works the same as the proposed one as Kinect recognized that finger based on depth and any open finger can be recognized and traced. If the user shows any stick, not his finger, then also it would be detected as finger. This false detection scenario gives an upper hand to vision based approach than sensor based. Kinect based system gives 100% correct results with segmentation (Raheja et al., 2011) but as Kinect is costlier in comparison with webcam and also it is a sensor, we stick to webcam for final results.

5. Conclusion

Currently paint drawing technology uses touch enabled screens or sensor gloves. In this paper natural-bare fingertip is used to draw visible content on screen in real time. It is like using

Table 1 Comparison of accuracy and related cost.

S. No.	Interaction method	Accuracy (%)	Devices needed and Issues
1	Gloves	100	Wired sensor gloves and user has to wear it ways. It is like using mouse. Sensor life is short
2	Touch enabled system (https://www.sensubrush.com)	100	Stylus and touch screen, life is short
3	Heat touch enabled system [tablets]	100	Finger works, but touch screen is costly
4	Hand gesture Bragatto et al. (2006), Araga et al. (2012) and Raheja et al. (2011b)	90–96	Webcam, depend on image background. 100% on plain background. Cheaper than sensor, life is longer

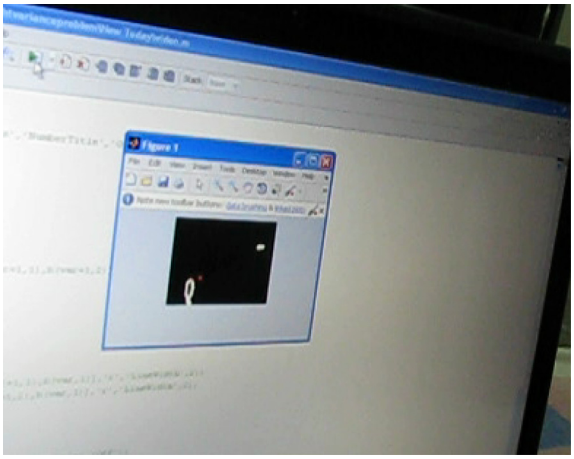

Figure 8 System setup.

stylus, where hand finger is doing that job. Many existing systems available for commercial and academic purposes like *Sensu* stylus[1], games (Hettiarchchi et al., 2013), touch enabled drawing[2,3](Buechley et al., 2009) etc. use sensor or touch screens but a vision based paint drawing where no sensor is used, would be an advantage to reduce the hardware cost and it is easy to use.

The proposed method provides a natural human–system interaction in such a way that it does not require a keypad, stylus, digital pen or glove for input. The method also shows the shape made by the user virtually. The shape would be shown on screen which the user has drawn in one session. This paper is an attempt to replace stylus with natural finger 'finger-stylus', where there is no need to touch the screen. In future, we would try to recognize these virtual shapes to bring them more useful recognitions and converting them into 3D for advanced applications. The system setup is shown in Fig. 8.

References

Araga, Y., Shirabayashi, M., Kaida, K., Hikawa, H., 2012. Real time gesture recognition system using posture classifier and Jordan recurrent neural network. In: IEEE World Congress on Computational Intelligence, Australia, pp. 1–8.

Berry, G., 1998. Small-wall, A Multimodal Human Computer Intelligent Interaction Test Bed with Applications (MS thesis). University of Illinois at Urbana-Champaign.

Booch, G., 2001. Artistic Vision: Automatic Digital Painting using Computer Vision Algorithms (thesis), University of Utah.

Bragatto, T.A.C., Ruas, G.I.S., Lamar, M.V., 2006. Real-time Video Based Finger Spelling Recognition System Using Low Computational Complexity Artificial Neural Networks, IEEE ITS, pp. 393–397.

Buechley, L., Hendrix, S., Eisenberg, M., 2009. Paints, paper, and programs: first steps toward the computational sketchbook. In: Proceedings of 3rd Int. Conf. on Tangible and Embedded Interaction, New York, USA, pp. 9–12.

Chaudhary, A., Gupta, A., 2011. Automated switching system for skin pixel segmentation in varied lighting. In: Proceedings of the 19th IEEE International Conference on Mechatronics and Machine Vision in Practice, Auckland, New Zealand, 28–30 Nov, 2012, pp. 26–31.

Chaudhary, A., Raheja, J.L., 2013. A health monitoring system for elder and sick persons. Int. J. Comput. Theory Eng. 5 (3), 428–431.

Chaudhary, A., Raheja, J.L., Das, K., Raheja, S., 2011a. A Survey on Hand Gesture Recognition in context of Soft Computing. In: Published as Book Chapter in Advanced Computing CCIS, vol. 133. Springer, Berlin Heidelberg, pp. 46–55.

Chaudhary, A., Raheja, J.L., sINGAL, K., Raheja, S., 2011b. An ANN based Approach to Calculate Robotic Fingers Positions. In: Published as Book Chapter in Advances in Computing and Communications, vol. 192. CCIS, Springer, pp. 488–496.

Cooper, H.M., 2012. Sign Language Recognition: Generalising to More Complex Corpora (Ph.D. thesis). University of Surrey, UK.

Forsline, L.B., Pedersen, B.D., 2004. Computer painting system with passive paint brush stylus US Patent no. US6801211 B2, Oct 2004.

Hettiarchchi, A., Nanayakkara, S., Yeo, K.P., Shilkrot, R., 2013. FingerDraw: more than a digital paintbrush. In: ACM SIGCHI Augmented Human, March, 2013, pp. 1–4.

Ju, S. Black, M., Minneman, S., Kimber, D., 1997. Analysis of gesture and action in technical talks for video indexing. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 595–601.

Liang, R. et al.,1998. A real time gesture recognition system for sign language. In: Proceedings of 3rd Intconf on FGR, USA, pp. 558–567.

Raheja, J.L., Chaudhary, A., Singal, K., 2011. Tracking of fingertips and centre of palm using KINECT. In: Proceedings of the 3rd IEEE International Conference on Computational Intelligence, Modelling and Simulation, Malaysia, 20–22 Sep, 2011, pp. 248–252.

Raheja, J.L., Das, K., Chaudhary, A., 2011a. Fingertip Detection: A Fast Method with Natural Hand. Int. J. Embedded Syst. Comput. Eng. 3 (2), 85–89.

Raheja, J.L., Manasa, M.B.L., Chaudhary, A., Raheja, S., 2011. ABHIVYAKTI: hand gesture recognition using orientation histogram in different light conditions. In: Proceedings of the 5th Indian International Conference on Artificial Intelligence, India, 14–16 Dec, 2011, pp. 1687–1698.

Raheja, J.L., Chaudhary, A., Maheshwari, S., 2014. Automatic Gesture Pointing Location Detection. Optik 125 (3), 993–996.

Rehg, J., Kanade, T., 1993. DigitEyes: Vision-Based Human Hand-Tracking, Technical Report CMU-CS-93-220, Carnegie Mellon University, December 1993.

Sato, Y., Kobayashi, Y., Koike, H., 2000. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In: Proceedings of IEEE Int. Conf. on Automatic FGR, pp. 462–467.

Starner, T., Pentland, A., 1995. Real time American Sign Language Recognition from Video using Hidden Markov Models, Technical Report 375, MIT Media Lab.

Sudderth, E.B., Mandel, M.I., Freeman, W.T., Willsky, A.S., 2004. Visual hand tracking using nonparametric belief propagation. In: IEEE CVPR Workshop on Generative Model Based Vision, 2004, pp. 1-9.

Wachs, J. et al, 2008. A gesture-based tool for sterile browsing of radiology images. J. Am. Med. Inform. Assoc. 15, 321–323. http://dx.doi.org/10.1197/jamia.M24.

Wang, R.Y., Popovíc, J., 2009. Real-Time Hand-Tracking with a Color Glove, ACM Transaction on Graphics, vol. 28, Issue 3, USA.

Wang, R., Paris, S., Popovic, J., 2011. Practical color-based motion capture. In: Eurographics/ACM SIGGRAPH Symposium on Computer Animation, pp. 139–146.

Zeller, M. et al., 1997. A visual computing environment for very large scale biomolecular modeling. In: Proc. IEEE Int. Conf. on Application specific Systems, Architectures and Processors (ASAP), Zurich, pp. 3–12.