Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, November 2015
ISSN 2074-8523

Fingertips Tracking based on Gradient Vector

Ahmad Yahya Dawod[1], Md Jan Nordin[1], and Junaidi Abdullah[2]

[1]Pattern Recognition Research Group,
Centre for Artificial Intelligence Technology (CAIT),
Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia, 43600 Bangi, Malaysia.
e-mail: ahmadyahyadawod@gmail.com, jan@ukm.edu.my
[2]Faculty of Information Technology,
Multimedia Universiti (MMU), 63000 Selangor, Malaysia.
e-mail: Junaidi2005@gmail.com

Abstract

This paper presents a new algorithm to detect and track fingertips position between frames using randomized circle detection. In order to reduce the processing time, a bounding box is created around the fingertips before the algorithm is applied. The randomized circle detection algorithm is employed based on the fact that for a circle, the gradient vector of the edge pixels will always point to the center of the circle. Therefore, if the gradient vector (which is coincident with the gradient line) for all the edge pixels around the circle are drawn, they will intersect at the center of the circle. Hence, the center of the circle can be determined by choosing the point that has the most number of gradient line intersections. The idea can be modified to be applied to semi-circle shape like fingers. Experimental results show good overall performance of individual fingertips tracking with average speed of less than 1s for each frame.

Keywords: *Fingertips Detection, Gradient Vector, Intersection Point, Edge Detection.*

1 Introduction

Fingertip tracking is an active area of research in the vision community. One of the important features of hand is the position of the fingertips being tracked. [1] Presented a method for multi-point gesture recognition based on machine vision. The experimental results showed that the method achieved good performance in fingertips tracking and multi-point gesture recognition. The method can achieve similar performance of multi-touch iPhone gesture without any expensive multi-touch device. [2] presented a method for tracking fingers in real time without the need to build the 3D model of the hand. The method extracted the fingers from the background through background subtraction. [3] proposed a vision-based fingertips tracking system. The system employed an infrared camera and a Kalman filter to detect and track the fingers. Using an infrared camera, image regions that match to human skin can be easily identified by transforming the image to grayscale image with a proper threshold value. The method used a normalized correlation template of a circle with the accurate size for fingertips detection. Then, the finger tracking system measures the trajectories of multiple fingertips by taking connections of fingertips detected in each image frame using Kalman filter.

In [4], a method that detects fingertips to reconstruct the six degree–of–freedom of the camera pose relative to the extended hand is introduced. A hand-pose model is created in a one–time calibration stage by calculating the fingertips locations in the presence of ground–truth scale information. The proposed method can stabilize 3D graphics annotation under different environment and variable fingertip size. In [5], a hand tracking method for stereo vision is proposed. The method used a real time tracking using two low–cost web cameras to track the 3D location and 2D orientation of the thumb and index finger of each hand for obtaining results of up to 8 degrees of freedom for each hand. The system estimated the movement of the tips locations based on the position of thumb and index finger of each hand. A calibrated pair of cameras is positioned to capture the image of the hands from above with the palms facing downward. The motivation for this hand tracker was a desktop-based two-handed interaction system in which the proposed method is able to select and manipulate 3D geometry in real-time using natural hand motion.

Sato *et al* [6] proposed a method for fingertips tracking under changing lighting conditions. This method is designed to have feature state-space estimation with uncertain model's algorithm, which attempts to control the influence of uncertain environmental conditions. The performance of the fingertips tracking system is enhanced by adapting the tracking model to compensate for the uncertainties inherent in the data. In [7], a method for fingertips position tracking in real time is presented. The system uses an infrared camera to segment between skin and non-skin pixels. This method tracks multiple fingertips reliably in cluttered background under changing lighting conditions. The method creates a fixed-size search window that works dependably because of the distance from the camera to hand region on the augmented desk interface system. The method computes the location of the search window via a morphological erosion operation of an extracted hand region.

In [8], a method for fingertips tracking using palm centers and fingertips for both left and right hands is presented. The method employs infrared camera images for reliable detection of human hands and uses a template matching technique for finding fingertips. Morphological operator is applied to the created shape of the palm of a human hand until the region develops into small blob at the center of the palm. In [9], a fingertip tracking technique by combining mean-shift and particle filter is presented. Morphological operators are applied to the segmented image to remove noise. Image gradients are calculated based on different scales. One of the problems of this method is that the error on the fingertip localization increases across time.

In [10], an algorithm for fingertips tracking using a skin detection algorithm is proposed. A curve estimated algorithm is used to smooth out the outline of the fingers and further analysis is done to identify each finger. The method finds the difference in terms of the arc tangent of the x, y and z for each finger to find how much each finger rotates and also translate in the x, y and z direction. In [11], fingertips tracking using a model namely the Fitts law is presented. The method employs a simple technique to locate a correctly oriented fingertip. The camera captures images and the desktop processes the image in order to segment the moving hand from the background image.

In [12], a method for fingertips are detected by combining three depth-based features and tracked with a particle filter over successive frames is presented. In this paper, a new and efficient technique for fingertips tracking based on randomized circular detection for each finger by the intersection points of gradient lines is proposed. The fingertips location in the first frame is found using reliable technique and this will serve as the starting point for tracking the fingertips in the remaining frames. To detect the subsequent fingertips location, a modified randomized circle detection algorithm is employed. The experimental results show acceptable tracking accuracy.

2 Proposed Method

The proposed method modifies the randomized circle detection method presented in [13]. The randomized circle detection algorithm is chosen based on the fact that for a circle, the gradient vector of the edge pixels will always point to the center of the circle. Therefore, if the gradient vector (which is coincident with the gradient line) for all the edge pixels around the circle are drawn, they will intersect at the center of the circle. Hence, the center of the circle can be determined by choosing the point that has the most number of gradient lines intersect. The intersection plane is considered as an integer plane, otherwise it is very difficult to get similar values for the intersection points. Based on this principle, since fingertips shape can be assumed to have a semi-circular shape, the algorithm can be employed with some modifications that suit the fingertip shape. The processes involved can be summarized in Fig. 1.

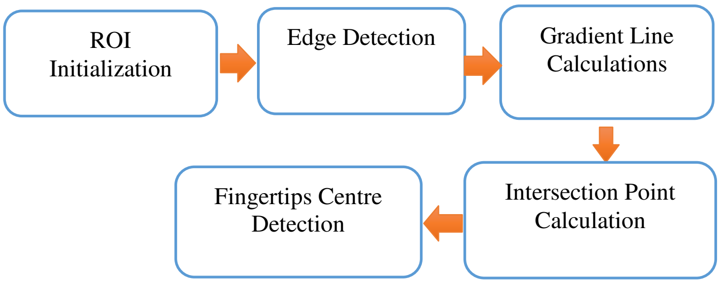
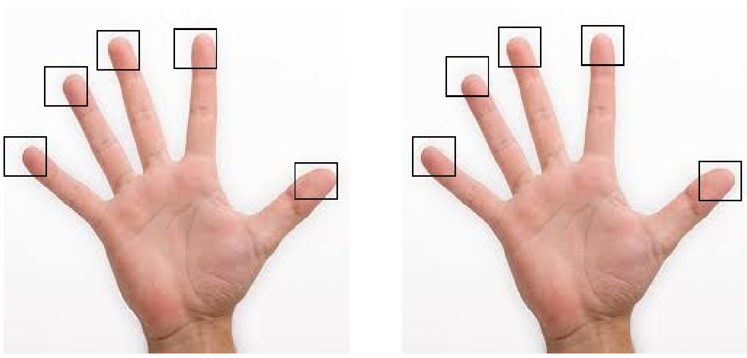


Fig. 1: Processes involved in tracking the fingertips in each frame.

2.1 ROI Initialization

In order to track the fingertips, the location of the fingertips in the first frame must be known in advance. In order to solve this, hand detection stage and then fingertips detection stage were proposed in the previous research work. Once the fingertips are detected, this will serve as the starting point to track the location of fingertips in the subsequent frames. ROI Initialization or Region of Interest Initialization module basically is a process of selecting few areas in the image to be focused for processing. This will increase the speed of tracking because instead of processing the whole image for the next subsequent frame, only few selected areas will be used to apply the randomized circle detection algorithm. In this case, it is assumed that there are 5 fingertips points that have been detected from the first image frame. Based on the detected fingertips, a boundary box of size 40×40 will be set as the initial ROI in which the center of the bounding box will be the location of the detected fingertips in the first frame. The 40×40 pixel size is chosen based on the assumption that the movement of the fingertips will not be that much between consecutive frames. The boundary box initialization for single fingertip is illustrated in Fig. 2 (a) and (b).

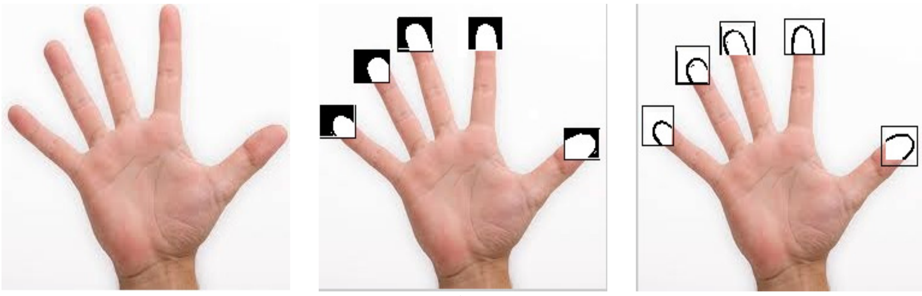


(a) (b)

Fig. 2: Boundary box initialization for single fingertip. (a) Previous frame, (b) Current frame

2.2 Edge Detection

Once the 5 ROIs have been determined in the current frame (assuming that 5 fingertips are detected), further processing will be done in the 5 ROIs. In this case, the canny edge detector is applied to obtain the edge map of the image edge Fig. 3 illustrates the detected edge using the Canny edge detector for a single fingertip.



(a) (b) (c)

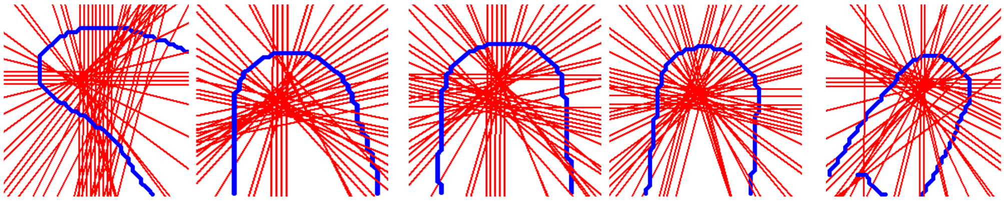
Fig. 3: Canny edge detector is applied to the five ROI, (a) Original fingertip, (b) Binary fingertips in the ROI, (c) Edge detection of the fingertips

2.3 Gradient Line Calculations

Given the edge map, the gradient lines are calculated for the fingertips edges. In this process, homogeneous distribution is used to choose edge pixels that will be part of the gradient line. Fig. 4 shows the resulting gradient lines drawn for each fingertips edge. On average, 40 to 60 gradient lines are found for each fingertip. For each region of interest, the boundary edge pixels are traced and the total number of edge pixels is recorded. If the number of edge pixels is 80 for example, only 40 gradient lines will be drawn. The algorithm will find the edge pixel that intersects with the edge of the ROI and start doing the tracing. The gradient line will be drawn starting from the edge pixel, p-start until p-stop where p-start is the mth edge pixel from the boundary while p-stop is the nth edge pixel after p-start. The value of m and n for the mth pixel and nth pixel are determined as follows:

$$m = \frac{totaledgespixels}{4} \quad (1)$$

$$n = \frac{totaledgespixels}{2} \quad (2)$$



Thumb Index finger Middle finger Ring finger Pinky

Fig. 4: Gradient Lines for five fingertips edges

2.4 Intersection Point Calculation

For each gradient line, the number of intersection points obtained will be tracked. If the number is greater than a certain threshold, the intersection points will be considered as the center of the curve. The followings describe the technique used to determine the intersection point of two lines (or line segments) in 2 dimensions.

The equations of the lines are

$$y = -x + 3 \tag{3}$$

$$y = 2x + 5 \tag{4}$$

To find the intersection, we equate Eq. (1) and Eq. (2) as follows:

$$-x + 3 = 2x + 5$$

$$x = -\frac{2}{3}$$

$$-2 = 3x \tag{5}$$

Substituting (2) into one of the Eq. (1) to find y

$$y = -x + 3$$

$$y = -\left(-\tfrac{2}{3}\right) + 3$$

$$y = \tfrac{11}{3} \tag{6}$$

The intersection found is illustrated in Fig. 5.

The point of intersection has an x-value between -1 and 0, y-value between 3 and 4.

The point of intersection is

$$(-\frac{2}{3},\frac{11}{3}) \approx (-1,4)$$

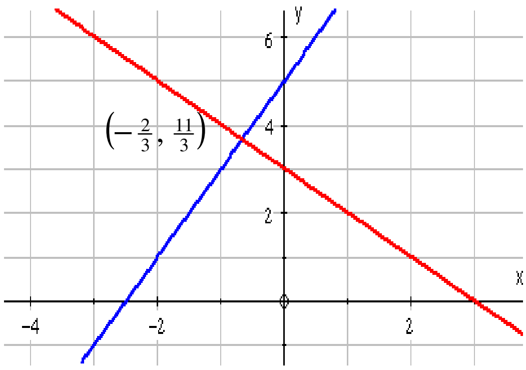


Fig. 5: The intersection point between two lines

2.5 Fingertips Centre Detection

The new center for fingertips tracking in the current frame will be based on the intersection point found from the above mentioned stages. The fingertips points are found from the intersection points that have the most occurrences. The algorithm for the above mentioned steps is summarized as follows:

For bounding box $B_i(i=1,2,3,4,5)$ for the five fingertips:

1. Obtain the edge map of the image using the Canny edge detector.

2. Obtain the number of curves, K, which consists of connected edge pixels with 8-adjacency connectivity (see Fig. 6).
   The 8-neighbours of pixels P are its vertical, horizontal and 4 diagonal neighbours denoted by $N_8(P)$.

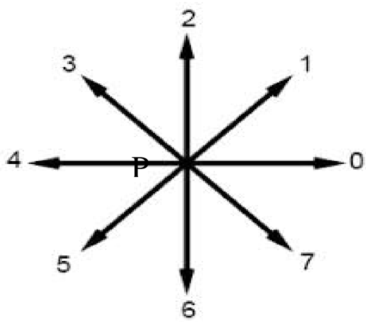


Fig. 6: Freeman's Chain Code

3. For connected curve $C_k\left(k=1,2,....,K\right)$:
   a. Create set V that consists of all the edge pixels of C_k.
   b. Calculate the number of edge pixels in V denoted as NUM_v.
   c. If $NUM_v > T_{\min}$, do ($T_{\min}$ is a threshold value)

   i. Obtain N edge pixels denoted as $p_j\left(j=1,2,...,N\right)$

   ii. Compute the gradient line of edge pixel p_j Which is denoted by $L_j(j=1,2,...,N)$

   iii. Find the intersection points and the point that is passed by the most lines is indicated by O. The number of intersections occurred at point O is denoted as O_{num}.

   In this case $T_{\min}$ is set to 20 based on [13].

3 Experimental Result

Time taken for each fingertip to be processed using the proposed method is shown in Table 1. The results were obtained using Matlab software. The number of frames involved is 30 frames.

Table 1: Results for fingertip speed testing

	Fingers	Average speed (Second)
1.	Thumb	0.044
2.	Index Finger	0.043
3.	Middle Finger	0.042
4.	Ring Finger	0.039
5.	Pinky	0.041

Results show that the time taken to run the algorithm can achieve less than 1s for 8G RAM, 1.6GHz (4 core). This can further be improved if gradient calculation is calculated in selective number of edge pixels rather than all of them. A threshold can also be set so that the gradient calculation will stop being calculated once the number of occurrences of intersection points has reached the threshold value which could not be done due to time constraints shown in Fig. 7 shows some of the results obtained.

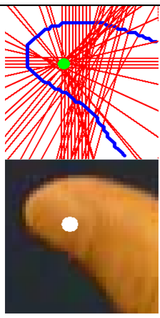
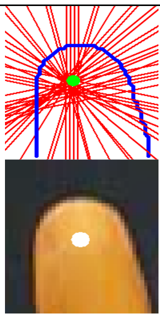
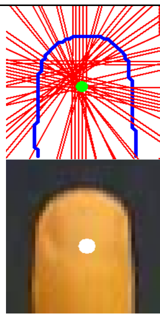
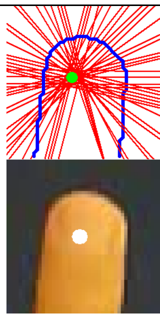
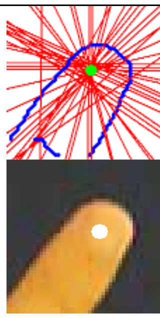
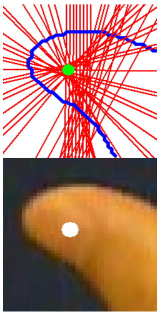
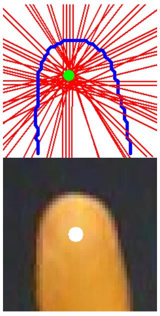
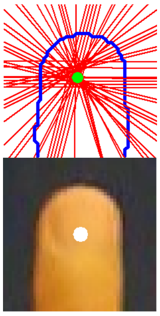
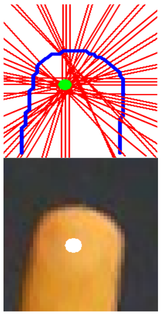
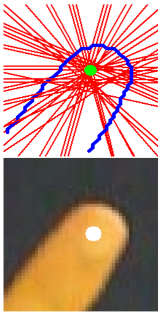
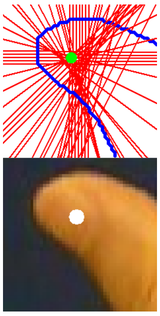
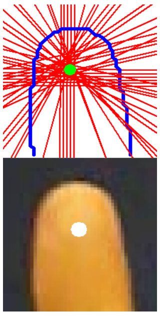
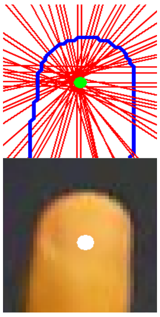
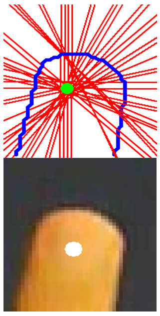
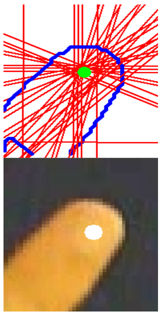
F. N	Thumb	Index finger	Middle finger	Ring finger	Pinky
1	 X = 20, Y = 22	 X = 25, Y = 27	 X = 27, Y = 27	 X = 24, Y = 21	 X = 22, Y = 25
10	 X = 22, Y = 23	 X = 24, Y = 21	 X = 24, Y = 23	 X = 27, Y = 18	 X = 23, Y = 25
20	 X = 18, Y = 22	 X = 22, Y = 20	 X = 26, Y = 27	 X = 31, Y = 22	 X= 23, Y= 22

Fig.7: The Detected Coordinate of the Intersection Points with the most Occurrences

4 Discussion for Fingertips Tracking

Experimental results show that after tested the proposed method using the randomize circle detection for each finger, fingertips positions are found accurately in each image frame. The processing speed of finding the fingertip positions is also quite reasonable. Instead of processing all the pixels in the ROI, only the edge pixels are considered. Based on experimental that use had done, it is found that the Canny edge detector method works better than any other well-known edge detector methods.

The size of the ROI, 40x40, is decided based on the experimental facts that the movement of the fingers in the video sequence is moderate but not very fast. In the fast movement of the fingers in video sequence a larger size can be considered. The detection accuracy and computation performance are two main concerned issues. The method is reasonably fast for five fingertips.

5 Conclusion

In this paper, a method for tracking fingertips by using modified randomized circular detection is proposed. The method starts with identifying region of interest from the detected fingertips in the previous frame, applying edge detection and calculating the gradient vector line. The fingertips centers are found based on the gradient line intersections that mostly occur. Region of interest is identified so that the method will focus on specific areas in the frame to be processed. The results obtained show good performance on the tracked fingertips in terms of accuracy and speed. Future works might include experimental works for closed fingers where the ROI is difficult to be extracted.

ACKNOWLEDGEMENTS
The researchers wish to thank Universiti Kebangsaan Malaysia (UKM) and Ministry of Education, Malaysia for supporting this work by research grant DIP-2014-018 and FRGS/1/2014/ICT07/UKM/02/2.

References

[1] Wensheng, L. and Huaiwen, H. Fingertip tracking and multi-point gesture recognition. in *Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on*, 2010, pp. 1-4.

[2] Shaker, N. and Abou Zliekha, M. Real-time finger tracking for interaction. in *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, 2007, pp. 141-145.

[3] Sato, Y., Oka, K., Koike, H., and Nakanishi, Y. Video-based tracking of user's motion for augmented desk interface. in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, 2004, pp. 805-809.

[4] Lee, T. and Hollerer, T. Handy AR: Markerless inspection of augmented reality objects using fingertip tracking. in *Wearable Computers, 2007 11th IEEE International Symposium on*, 2007, pp. 83-90.

[5] Malik, S. 2003. Real-time hand tracking and finger tracking for interaction CSC2503F project report. *Department of Computer Science, University of Toronto, Tech. Rep,*

[6] Dominguez, S. M., Keaton, T., and Sayed, A. H. 2006. A robust finger tracking method for multimodal wearable computer interfacing. *Multimedia, IEEE Transactions on,* vol. 8, pp. 956-972.
[7] Oka, K., Sato, Y., and Koike, H. Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, 2002, pp. 429-434.
[8] Sato, Y., Kobayashi, Y., and Koike, H. Fast tracking of hands and fingertips in infrared images for augmented desk interface. in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 462-467.
[9] Do, M., Asfour, T., and Dillmann, R. Particle filter-based fingertip tracking with circular hough transform features, 2011. in *Proc. of the 12th IAPR Conf. on Machine Vision Applications*, 2011.
[10]Patil, M. P. M. and Patil, Y. Robust Skin Colour Detection And Tracking Algorithm. in *International Journal of Engineering Research and Technology*, 2012.
[11]Brown, T. and Thomas, R. C. Finger tracking for the digital desk. in *User Interface Conference, 2000. AUIC 2000. First Australasian*, 2000, pp. 11-16.
[12]Liang, H., Yuan, J., and Thalmann, D. 3D fingertip and palm tracking in depth image sequences. in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 785-788.
[13]Jia, L.-q., Peng, C.-z., Liu, H.-M., and Wang, Z.-H. A fast randomized circle detection algorithm. in *Image and Signal Processing (CISP), 2011 4th International Congress on*, 2011, pp. 820-823.