2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.2.26

Real-Time Stereo Vision Based Fingertip Detection and Tracking

Xia Yuan[+] and Qunsheng Peng

State Key Lab. of CAD&CG, Zhejiang University

Hangzhou, Zhejiang, PR China

Abstract—Fingertip detection and tracking is the foundation of many hand-based natural Human Computer Interaction (HCI) applications. Previous methods either can hardly provide a natural way of interaction, or are too complex to be adopted in practical systems. In this paper, we propose a real-time stereo vision-based fingertip detection and tracking method that works very well in real environment. Our method can get the accurate fingertip position regardless the finger shape for the single finger. Fingertip detection is formulated as a graph theory problem, and can be solved easily given the 2D finger skeleton image. The accurate 3D location of the fingertip can be obtained based on stereo vision. To achieve the robustness against noise and occlusion, Kalman filter is applied to smooth the trajectory of fingertip. Our method is simple yet effective, and can run in real-time on a normal PC.

Keywords-fingertip detection; finger tracking; stereo vision; finger segmentation; finger skeleton; trajectory; Kalman filter

1. Introduction

In recent years, fingertip detection and tracking attracts more and more attention in the research field of hand-based natural Human-Computer Interaction (HCI). Especially for the single fingertip detection and tracking, it has been widely studied in hand pose recognition, virtual mouse, and applications of augmented reality (AR). A large number of methods about fingertip detection and tracking have been proposed.

Data-Glove based approaches [1],[2],[3] capture hand movement information by wearing special marked gloves. In [4], J. Zeng et al. introduced a color feature-based fingertip detection method, which dyes the fingertip with different colors as the feature. These methods can track hand and detect fingertip accurately, but it is not a natural way of interaction and the hardware is costly. Model-based fingertip detection and tracking has also been widely studied. In [5], I. Kazuyuki designed a hand tracking system which is based on skin color model of hand. However, in real environment, this method may fail frequently due to the wide variety of lighting condition. In general, fingertip model includes 3D fingertip models [6] and 2D fingertip models [7]. Since large computational cost is involved in these methods, they are not suitable to be adopted in real-time systems. Feature-based fingertip detection methods are also researched widely. D. Yang et al in [8] proposed a match method based on fingertip ring feature. Once the fingertip feature is occluded, fingertip position can not be found and finger tracking would fail. In addition, improper assumptions are given in some methods in order to reduce the complexity, e.g. finger is always straight or fingertip is upward in movement. These methods are of great limitation in practical applications.

In recent years, due to the rapid development of computer vision, vision-based hand tracking and fingertip detection is becoming more and more popular. In [9], Z. Zhang et al. employed an arbitrary quadrangle-shaped panel and tip pointer as visual panel to map the fingertip 3D position to the corresponding position on the display. In [10], E. Ali et al. presented a literature review on the latest finger research in this field. The

[+] Corresponding author.
E-mail address: yuanxia@zjucadcg.cn

main difficulties appearing in the vision-based hand tracking include, (1) Self-occlusions: hand projection in 2D image may result in a large number of shapes with serious self-occlusion, making it difficult to segment and to extract high level features. (2) Complex environments: the complexity of the background takes great influence on the accuracy of hand tracking. It is a great challenge to operate the hand-based HCI system with unrestricted backgrounds and changing lighting condition in real environments. (3) Processing speed: most of existing methods take large computational cost, which makes them hard to be adopted in real-time environment.

To overcome the above limitations of existing methods, this paper presents an approach of real-time stereo vision based fingertip detection and tracking. A simple yet effective approach is first applied to get the 2D position of the single fingertip, and then stereo vision information is incorporated to obtain the accurate 3D position. The main contributions of this paper include: (1) Finger Extraction: a classical background subtraction method is presented to extract finger real-timely in static background. (2) Fingertip Detection: after analyzing the drawbacks of existing methods, we propose a novel method which formulates fingertip detection problem as a graph theory problem. Our method can detect the accurate 3D position of single fingertip regardless the finger shape, even if finger is bended. It is simple yet effective, and can reach a speed of 15fps for 320×240 size of binocular video. (3) Finger Tracking: an approximate Kalman filter-based algorithm to predict the fingertip position and for tracking. A curve fitting method is proposed to render the trajectory of finger in order for good visual effect.

In experiment, to validate the proposed method, we apply our method to some applications for hand-based human computer interaction. Experimental results convince us that the proposed method can be adopted in real-time systems.

2. Finger Extraction

Our method is based on the stereo vision and the parameters of the two cameras are fixed. We can obtain 3D space location information by camera calibration technology [11], and our attention focus on a virtual cube as the region of interest, which is rendered before finger extraction. Therefore, in the rest of the paper we will assume that the cameras are calibrated and the virtual cube has been rendered in the real scene. Meanwhile, the 3D space location information of virtual cube is also known. Fig.1 shows a virtual cube as the region of interests in the real 3D scene.

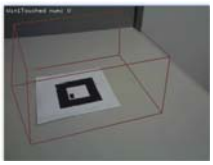


Fig. 1. Virtual cube: it is as the region of interests for research of this paper

Once the fixed cameras have been calibrated, the background is recorded. To extract moving finger from the static background, we employ the classical background subtraction to segment the finger. The algorithm includes mainly two key parts as follows:

2.1. Finger Segmentation

A non-parametric estimation method is applied for background modeling, which is proposed by Elgammal and David [12]. The background Probability Density Function (PDF) is given by the histogram of the n most recent pixels in the image, which is smoothed with Gaussian kernel. The PDF of pixel x is denoted by $pdf(x)$. If $pdf(x) > th$, the pixel x is classified as background, where th is a global threshold though the whole image.

Native description of the approach: detect the foreground objects according to the difference between the current frame and the background image:

$$\left| frame_i - background_i \right| > T \tag{1}$$

where T is a threshold based on the whole image. According to the above equation, the foreground finger can be extracted easily.

2.2. Background Updating

For a new pixel sample, we update the background by the follow principle: *Selective Update:* add the new sample to the model only if it is classified as a background sample.

Binary segmentation of single finger can be got by image binaryzation. Fig.2 (b) shows final binary results of single finger extraction from video.

(a) (b) (c)

Fig.2. Binary segmentation and skeleton image of single finger. (a) source image. (b) finger segmentation image. (c) finger skeleton image

3. Finger Detection

The extracted finger is represented as a binary image of finger in 2D, and then we detect the fingertip position from the image. Our algorithm includes two key steps:

3.1. Finger Skeleton Extraction

Note that the position of fingertip must be on the axis of the single finger, that is, the finger skeleton. Therefore, we can remove the redundant pixels of the finger, which deviate from the skeleton. In this way, we can detect fingertip in a smaller pixel set, which improves both efficiency and accuracy.

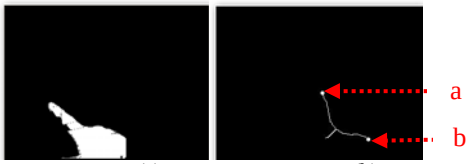
Finger skeleton can be extracted by using thinning and skeleton algorithm [13], which is proposed by T. Y. Zhang and C. Y. Suen. Generally speaking, there may be a few noise pixels in the finger skeleton image, which can be removed by regional connectivity analysis. Fig.2 (c) shows the result of finger skeleton.

3.2. Fingertip detection

1) Candidate points for 2D fingertip

For fingertip detection of single finger, it is obvious that the fingertip position must be one of the endpoints on the finger skeleton, as shown in Fig.3. Hence, the problem of fingertip detection is converted to finding the longest path among all the shortest paths connecting pairs of endpoints, and then the two endpoints of the path are the two candidate points of the fingertip.

Fingertip detection described above can be regarded as a path search problem in graph theory. We need to find a path which meets two conditions: 1. the path is the shortest path between two endpoints. 2. the path is the longest among all shortest paths in the finger skeleton image.



(a) (b)

Fig.3. (a) finger segmatation image. (b) finger skeleton image. a,b are two candidate points of fingertip positions,they are two endpoints of longest path in finger skeleton image.

Firstly, the binary image of finger skeleton can be regarded as a graph. Considering the binary image of finger skeleton as an undirected graph $G<V,E>$, V is the set of vertices, E is the set of edges. Here, vertices are all pixels, and edges are the relations of any pairs of pixels. It is a 2D graph with $W\times H$. W is width, H is height of image. Let Ω be the set of pixels which are on the finger skeleton, $\omega(x,y)$ is the weights function of edges. The value of pixel x is denoted by $v(x)$, $v(x)=1$ if and only if the pixel x is on the finger skeleton. The graph of finger skeleton image in 2D can be expressed as:

$$v(x)=\begin{cases}1 & if\ x\in\Omega\ \ valid\ \ pixel\\ 0 & if\ x\notin\Omega\ \ invalid\ \ pixel\end{cases} \tag{2}$$

Secondly, the shortest path problem will be discussed only for these valid pixels which are denoted by $v(x)=1$ in (2). For each pixel $x \in \Omega$, it can arrive directly to pixel y, one of its eight neighborhood pixels if and only if the pixel $y \in \Omega$. After that, we can get the weights of edges in graph as follows:

$$\omega(x,y)=\begin{cases}1 & if\ y\in\Phi_e\ and\ y\in\Omega \\ \infty & else\end{cases} \tag{3}$$

Where Φ_e is the set of eight neighborhood pixels for a pixel. Endpoints are expressed as at least one of its eight neighborhood pixels (Φ_e) $x \notin \Omega$, and Γ is the set of endpoints.

From above definitions, any classical shortest path algorithm can be used to solve the problem under the constraint of the condition 1. To find all shortest paths among pairs of endpoints, we need to solve the shortest paths among multi-source vertices. Bellman-Ford algorithm, SPFA algorithm etc. are classical algorithms to solve the problem. After analyzing performance of these algorithms, for 320×240 size of input videos, we apply Dijkstra algorithm [14] to find the shortest paths among all pairs of endpoints in the graph. The performance of algorithm is efficient, and can reach 15fps in real-time.

Once the longest path is determined, the two endpoints of the longest path are then taken as the candidate points of fingertip position in 2D image.

2) Accurate 3D fingertip position

The purpose of 3D reconstruction is to restore the 3D position of the fingertip. 3D reconstruction based on stereo vision has been studied much. In [11], S. D. Ma proposed an effective method for 3D reconstruction. We adopt this method to reconstruct the two candidate points for finger position in our paper.

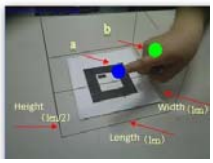
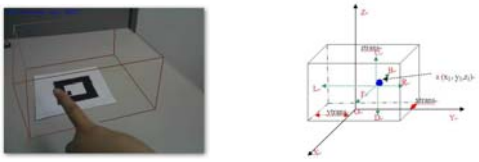


Fig.4. Two 3D candidate points of fingertip position. (a) blue. (b) green.

To distinguish the accurate fingertip from the two 3D candidate points of fingertip for single finger, virtual cube (Fig.1) is considered to solve the problem. As is mentioned in section 2, a virtual cube has been drawn as the region of interests, and its 3D location is known after camera calibration. Therefore, we can unify the virtual cube and finger in the real environment. As shown in Fig.5 (a), it is obvious that finger always intersects with the cube, and the intersection point is not fingertip between the two candidate points of fingertip position, and should be discarded. Denote by a, b the two candidate points of the fingertip position. The steps are as follows:



(a) (b)

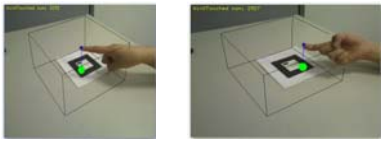
Fig.5. (a) Finger always interacts with cube when it enters into the interesting region. (b) Computing distance of point to six surfaces.

Step1: compute respectively the distance from points to six sides of the cube (Fig5. (b)).

Step2: sorting all distance values and discarding the point whose distance is minimal.

Step3: the other point is the true fingertip.

The proposed finger detection method is robust because it is based on geometry information of finger graph directly. As shown in Fig.6 (bended and straight fingertip), the proposed method can compute accurately the fingertip position regardless the finger is straight or not. Meanwhile, our algorithm is efficient and can detect fingertip accurately in real-time.



(a) (b)

Fig.6. Finger detection in different finger shape. (a) straight finger. (b) bended finger.

4. finger tracking

4.1. Tracking Finger

It is very important to track moving finger for analyzing finger movement, which is useful in HCI systems.

The fingertip obtained with the above method is not stable due to noise and occlusion. On one hand, the finger may be occluded, in which case error of the fingertip position is inevitable. On the other hand, finger movement is continuous and predictable. Therefore, we need to process the initial fingertip further in order for a stable result.

We propose a simple yet effective Kalman filter-based method [15] to solve the above problem. According to the history information of fingertip position, the reasonable fingertip positions in the current frame can be predicted. There are three important values in Kalman filter algorithm: measured value, estimated value and correct value, where the initial fingertip position is measure value, estimated value is predicted by computing finger speed in moving, the correct value is the result, and can be obtained by combing the measured and estimated value. Generally speaking, we can predict the accurate result by fingertip positions in previous 3-4 frames. Fig.7 shows the three values of one frame.

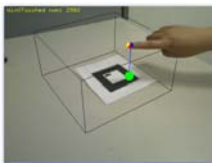


Fig.7. igFinger detection by kalman filter with three values: measure value(blue), estimate value (yellow) and correct value (red)

4.2. Trajectory of finger

Trajectory of the finger can provide us with its movement information. So it is necessary to render correct the trajectory of finger movement. However, rendering the trajectory directly according to fingertip positions cannot result in a smooth curve. Assuming that the ideal finger trajectory is smooth, and can truly reflect finger movement, the trajectory then can be represented as Non-Uniform Rational B-Splines (NURBS), and can be fitted easily with the method in [16]. Fig. 8. shows the final trajectory of finger movement.

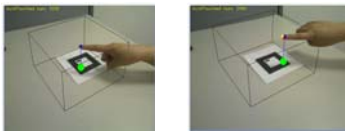
Fig.8. The trajectory of finger moving

5. Experimental Results

All experiments in our framework are conducted on a desktop PC with Intel ® Xenon ® CPU E5540@2.83GHz and 2.0G RAM, the two cameras are both Logitech QuickCam S5500 with 1.3 mega pixels, with maximum frame rate of 30 fps. The environment is an indoor scene with fixed cameras and consistent lighting scene.

Fig. 9 shows the procedures of fingertip detection from finger extraction to fingertip localization. In order to evaluate our method intuitively, we display the fingertip position and the virtual cube real-timely in 3D space. The success rate of our method is about 90%, and errors are mainly caused by fast movement and disturbance in the environment.

(a) (b) (c)



(d) (e)

Fig.9. The procedures of finger detection and finger tracking in our method. (a) finger enter into the interesting region. (b) finger segmentation result. (c) finger skeleton . (d) the initial fingertip position(blue point). (e) predict the accurate fingertip position by Kalman filter

Our method can be used for HCI in various ways. We test our method with two practical applications in AR. Fig.11 shows the virtual draw with finger in 3D scene. User can draw lines, simple flowers, write simple words continuously. Fig.10 (a) shows a drawing, and Fig.10 (b) shows a simple word.

The method can also be applied to games so that it can be controlled easily with finger. As shown in Fig.10 (c), we control movements of virtual fireball in terrain scene.

(a) (b) (c)

Fig.10. Drawing with finger. (a) drawing. (b) conrolling the hot fall. (c) writing some simple words

6. Conclusion

In this paper we proposed a method of stereo vision based single fingertip detection and tracking. Unlike existing methods, our method formulates the single fingertip detection problem as a graph theory problem. It is simple yet robust, even if the finger is bended, and can run in real-time by frame rate of more than 15 fps. To eliminate disturbance caused by noise and occlusion, Kalman filter is applied to smooth the trajectory. Compared with previous methods, the resulted fingertip position is more accurate and more stable.

The limitation of our method is that it considers only the case of single fingertip. Multi-fingertip detection has more attractive applications in HCI. Therefore, we plan to consider multi-fingertips detection in the future work. In addition, the speed of fingertip detection may become the bottleneck of performance for large size of input videos, so a fast implementation on GPU is also worth to be considered.

7. Acknowledgment

This paper is supported by 973 program of china (No. 2009CB320802).

8. Reference

[1]. D.J. Sturman, D. Zeltzerm, “A Survey of Glove-based Input,” IEEE Computer Graphics and Applications, Vol.14, No.1, Jan.1994, pp. 30-39

[2]. B. H. Thomas, W. Piekarski, “Glove based user interaction techniques for augmented reality in an outdoor environment,” Virtual Reality: Research, Development, and Applications, Vol.6, No.3, 2002, pp.167-180.

[3]. F. Parvini, D. McLeod, “An Approach to Glove-Based Gesture Recognition,” Lecture Notes on Computer Science, Vol.5611 2009, pp. 236-245

[4]. Jianchao Zeng, Yue Wang, M. Freedman, “Color-Feature Finger Tracking of Breast Palpation Qnatification,” IEEE International Conference Robotics and Automation Alquerqu. New Mexico IEEE Computer Society, 1997, pp.2565-2570.

[5]. K. Imagawa, S. Lu, S. Ig, “Color-Based Hands Tracking System for Sign Language Recognition,” Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp.462

[6]. D. James, S. Mubarak, “Determining 3-D Hand Motion,” Proc.28th Annual Asilomar Conference on Signals, Systems and Computers, Pacific Grove:IEEE Computer Society, 1994, pp. l12-120

[7]. K. Oka, Y. Sato, H. Koike, “Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems,” The Fifth IEEE International Conference on Automatic Face and Gesture Recognition, Washington D.C: IEEE Computer Society , 2002, pp.429-434.

[8]. D. D. Yang, L.W. Jin, “Fingertip Detection Approach for Finger-Writing Chinese Character Recognition System,” Journal of south China university technology (Natural Science Edition) , 2007, Vol.35, No.1

[9]. Z. Zhang, Y. Wu, Y. Shan et al, “Visual Panel, Virtual Mouse, Keyboard, and 3D Controller with an Ordinary Piece of Paper,” ACM Workshop on Perceptive User Interfaces, 2001, pp.219-226

[10]. Erol, G. Bebis, “Vision-based hand pose estimation: A review,” Computer Vision and Image Understanding, Vol.108, No.1~2, Oct.2007, pp.52~73

[11]. D.M. Song, Y.Z. Zheng, “Computer Vision,” Science Press, 1998, pp.54~80.

[12]. Haritaol, D. Harwood, “non-parametric estimators for background model”

[13]. T. Y. Zhang, C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” Comm. ACM, 1984, vol.27, no.3, pp.236-239

[14]. W. Mark, “Data Structures and Algorithm Analysis in C (Second Edition),” Addison-Wesley, 1997, pp:224-229,294-295

[15]. W. Greg, B. Gary “An Introduction to the Kalman Filter,” University of North Carolina at Chapel Hill,1995

[16]. S. Richard, Jr. Wright, “OpenGL Super Bible 4th Edition,” Addison Wesley, 2007, pp.306-326