Robust Fingertip Tracking with Improved Kalman Filter

Chunyang Wang and Bo Yuan

Intelligent Computing Lab, Division of Informatics
Graduate School at Shenzhen, Tsinghua University
Shenzhen 518055, P.R. China
tsinglong2011@gmail.com, yuanb@sz.tsinghua.edu.cn

Abstract. This paper presents a novel approach to reliably tracking multiple fingertips simultaneously using a single optical camera. The proposed technique uses the skin color model to extract the hand region and identifies fingertips via curvature detection. It can remove different types of interfering points through the cross product of vectors and the distance transform. Finally, an improved Kalman filter is employed to predict the locations of fingertips in the current image frame and this information is exploited to associate fingertips with those in the previous image frame to build a complete trajectory. Experimental results show that this method can achieve robust continuous fingertip tracking in a real-time manner.

Keywords: multi-fingertip tracking, curvature detection, distance transform, Kalman filter, data association.

1 Introduction

In traditional human-computer interaction (HCI), users rely on devices such as keyboards, mice, remote controllers and touch screens as the control interface, causing various levels of inconvenience. With the popularization of smart TVs and smart phones, the desire for simple, natural and intuitive HCI techniques has been consistently increasing. Vision based HCI, which uses video to capture a user's information such as face, gait and gesture, has the inherent advantage of being natural and user friendly and has become a fast growing research area in recent years [1].

As a key component of vision based HCI, fingertip tracking features extensive application prospects. Previously, fingertip tracking has been realized using infrared camera [2], multiple cameras [3] and LED lights for marking fingers [4]. Although these methods did achieve fingertip tracking to some extent, they suffer from a number of issues: i) the use expensive cameras; ii) strict restrictions on the positions of fingers; iii) the need of auxiliary devices. By contrast, in this paper, we focus on real-time fingertip tracking using a single low cost optical camera and bare hands to improve the applicability and user experience.

For multi-target tracking, data association is the most commonly used method [5]. The typical methods of data association include nearest neighbor, probabilistic data association filter and joint probabilistic data association filter. A simple approach to multi-fingertip tracking is by associating the detected fingertips in the current image frame with the corresponding nearest fingertips in the previous image frame [6]. However, it is prone to false association when fingertips move quickly. In order to solve the mutual interference and the interfering noise in multi-fingertip tracking and make full use of the motion information of fingertips, Particle filter and Kalman filter have been used to improve the effectiveness of fingertip tracking. For example, the K-means clustering algorithm can be combined with Particle filter to solve the problem of mutual interference [7]. Although this method can track multiple fingertips accurately, the cost of computation is significant and is not suitable for real-time applications. In the meantime, Kalman filter can be used to predict the locations of fingertips in the new image frame and associate fingertips between image frames using the predicted locations [2]. This method needs to measure the time interval between image frames and the velocity of fingertips and cannot properly handle the sudden change of locations caused by the acceleration of fingertips.

In this paper, we employ an improved Kalman filter to predict the locations of fingertips under the assumption of uniform acceleration instead of uniform speed. In addition, we propose a method using distance transform to remove interfering points. In the rest of this paper, Section 2 describes the procedure of extracting the hand region using the skin color model. Section 3 shows how to detect fingertips by curvature and remove interfering points using the cross product of vectors and distance transform. Section 4 gives the details of our approach to multi-fingertip tracking. Section 5 presents the main experimental results and analysis and this paper is concluded in Section 6.

2 Hand Extraction

The purpose of hand extraction is to separate hand region from background. To extract hand efficiently, we use the skin color model. In computer vision, the color space includes RGB, YCbCr, HSV, YIQ and so on. According to the distribution of human skin color [8] and previous research results [9–11], YCbCr is selected in our work. The conversion from RGB to YCbCr is as below:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1686 & -0.3311 & 0.4997 \\ 0.4998 & -0.4185 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

In YCbCr color space, Y represents illumination while Cb and Cr represent chrominance information. Each image is segmented into skin region and non-skin region based on the values of Cb and Cr. The threshold values of Cb and Cr are set as below according to previous studies:

$$\begin{cases} 77 \le \mathrm{Cb} \le 127 \\ 133 \le \mathrm{Cr} \le 173 \end{cases} \tag{2}$$

The original color image in a real-world environment is shown in Fig. 1(a) and the binary image as the result of skin color segmentation is shown in Fig. 1(b). It is easy to see that there are some noise and holes in the initial binary image. Next, morphological operation with a 5-by-5 structuring element is applied to remove small noise and the morphological closing operation is used to remove small holes. After morphological operation, we calculate the size of each connected region and the largest one is identified as the hand region, as shown in Fig. 1(c). Note that, in normal circumstances, the hand is much closer to the camera than the human face and it is reliable to use area information to remove the face region.

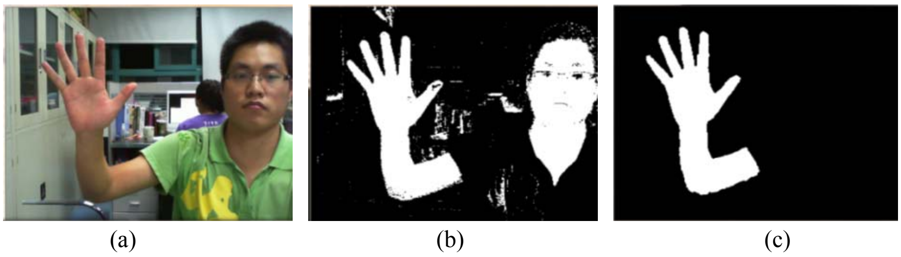


Fig. 1. Hand extraction: (a) Original image; (b) Skin color segmentation; (c) Binary hand image

3 Fingertip Detection

3.1 Curvature Detection

In order to detect fingertips from the contour of hand, the curvature-based algorithm is adopted [12–14]. The curvature of a contour point is represented by the cosine value of θ, which is the angle between $\overrightarrow{P_iP_{i\text{-}N}}$ and $\overrightarrow{P_iP_{i+N}}$:

$$\cos\theta = \frac{\overrightarrow{P_iP_{i-N}} \bullet \overrightarrow{P_iP_{i+N}}}{\left\|\overrightarrow{P_iP_{i-N}}\right\| \left\|\overrightarrow{P_iP_{i+N}}\right\|} \tag{3}$$

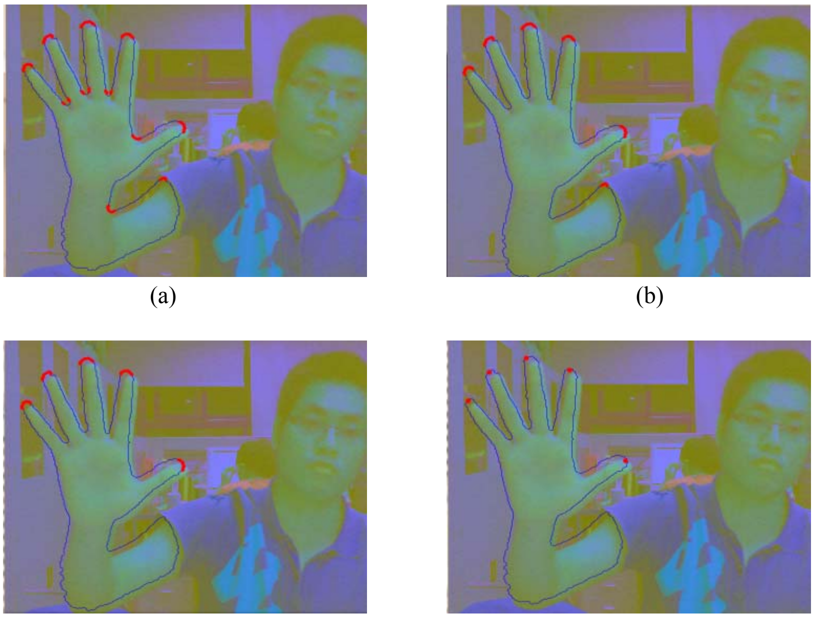
where P_i, $P_{i\text{-}N}$ and P_{i+N} are the i^{th}, the $(i\text{-}N)^{\text{th}}$ and the $(i+N)^{\text{th}}$ points in the contour, respectively. In practice, values of N between 5 and 25 work reasonably well.

Points with curvature values satisfying a predefined threshold value (close to 0) are selected as candidates for fingertips. An example of candidate fingertips after curvature detection is shown in Fig. 2(a). Note that the curvature values are also close to 0 for points located at places such as the valleys between fingers, the joint of arm and other peaks on the contour. These are regarded as interfering points and, in order to avoid false positive results, some measures should be taken to remove these points.

3.2 Fingertip Filtering

For points located at the valleys of the contour, the values of the cross product of vectors are different to those located at peaks (opposite sign). As a result, we can use this direction information to remove interfering points located at places such as the valleys between fingers and the joint of arm. The remaining candidate fingertips after being filtered by this method are shown in Fig. 2(b).

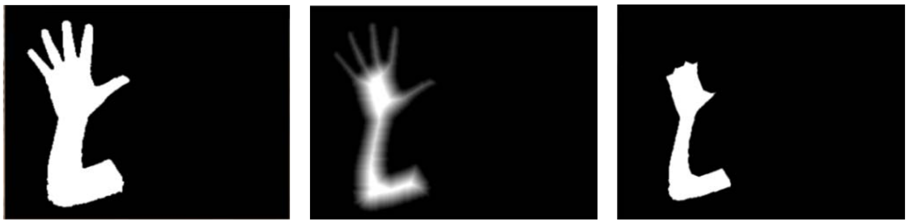
For interfering points located at the peaks of the contour, we can use distance transform to remove them [15].The result of distance transform on a binary image is a grayscale image as shown in Fig. 3(b), which will be further converted to a binary image. From Fig. 3(c), we can see that fingers are removed while the palm and arm are preserved by distance transform. For each candidate fingertip, we calculate its minimum distance to the contour and remove those candidates whose minimum distances are less than a threshold value. The candidate fingertips after distance transform are shown in Fig. 2(c). Finally, the locations of fingertips are obtained by clustering the remaining candidate fingertips, as shown in Fig. 2(d).



(a) (b)

(c) (d)

Fig. 2. Fingertip detection: (a) Curvature detection; (b) Cross product of vectors; (c) Distance transform; (d) Cluster analysis



(a) (b) (c)

Fig. 3. Distance transform: (a) Original binary image; (b) Gray image after distance transform; (c) Binary image after distance transform

4 Fingertip Tracking

For the locations X_k of fingertips in the k^{th} image frame, an improved Kalman filter is used to predict the locations $\hat{X}_{k+1}$ of fingertips in the $(k+1)^{\mathrm{th}}$ image frame. Then we use the nearest neighbor rule to associate fingertips between frames by comparing the predicted locations $\hat{X}_{k+1}$ with the detected locations X_{k+1} in the $(k+1)^{\mathrm{th}}$ image frame to achieve the robust tracking of multiple fingertips.

4.1 Fingertip Prediction

We take into account the location, velocity, and acceleration information of fingertips and use an improved Kalman filter to predict the locations of fingertips [2, 16–18].

The location, velocity and acceleration of a fingertip along a certain coordinate in the k^{th} image frame are represented by x_k, v_k and a_k, respectively. The kinematic equations describing the motion of a fingertip are shown as follows:

$$x_k = x_{k-1} + v_{k-1}T + a_{k-1}T^2/2 \tag{4}$$

$$x_{k\text{-}1} = x_{k-2} + v_{k-2}T + a_{k-2}T^2/2 \tag{5}$$

$$x_{k\text{-}2} = x_{k-3} + v_{k-3}T + a_{k-3}T^2/2 \tag{6}$$

$$v_{k-1} = v_{k-2} + a_{k-2}T \tag{7}$$

$$v_{k-2} = v_{k-3} + a_{k-3}T \tag{8}$$

Since the time interval T between two consecutive image frames is very short, we assume that the acceleration of fingertips in three successive image frames is approximately constant. Therefore, the following equation is obtained by combining (4), (5), (6), (7) and (8):

$$x_k = 3x_{k-1} - 3x_{k-2} + x_{k-3} \tag{9}$$

Note that velocity, acceleration and time interval between image frames are all eliminated in (9). With this simplified representation, it is now possible to only rely on the locations of fingertips in the previous three image frames to predict the locations of fingertips in the current image frame.

We define the state vector x_k as the locations of a fingertip along one coordinate axis in three successive image frames, $\mathrm{x}_k=[x_k, x_{k\text{-}1}, x_{k\text{-}2}]^{\mathrm{T}}$, and the observation vector z_k as the locations of the fingertip in the k^{th} image frame ($\mathrm{z}_k=x_k$).

The state and observation equations of system are:

$$\mathrm{x}_{k+1} = F_k \mathrm{x}_k + \Gamma_k \mathrm{w}_k \tag{10}$$

$$\mathrm{z}_k = H_k \mathrm{x}_k + \mathrm{v}_k \tag{11}$$

In the above, F_k is the state transition matrix; Γ_k is the noise matrix of system; H_k is the observation matrix; w_k is system noise, which influences the state of system; v_k is observation noise, which is the error between real and detected locations of fingertip.

The definitions of F_k, Γ_k and H_k are as follows:

$$F_k = \begin{bmatrix} 3 & -3 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{12}$$

$$\Gamma_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \tag{13}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{14}$$

Assume that w_k and v_k are the Gaussian white noise with zero mean and the covariance matrix of w_k and v_k are Q_k and R_k, respectively. In addition, w_k and v_k are independent of each other and also independent of the initial state of system.

The optimal predicted value and the covariance matrix of predicted error are:

$$\hat{\mathrm{x}}_{k+1|k} = F_k \hat{\mathrm{x}}_{k|k} \tag{15}$$

$$\mathrm{P}_{k+1|k} = F_k \mathrm{P}_{k|k} F_k^T + \Gamma_k \mathrm{Q}_\mathrm{k} \Gamma_k^T \tag{16}$$

In the above, $\hat{X}_{k+1|k}$ is the optimal predicted value in the (k+1)th image frame; $\hat{X}_{k|k}$ is the optimal estimated value in the kth image frame; $\mathrm{P}_{\mathrm{k+1|k}}$ is the covariance matrix of $\hat{X}_{k+1|k}$; $\mathrm{P}_{\mathrm{k|k}}$ is the covariance matrix of $\hat{X}_{k|k}$.

After obtaining the observation vector, the optimal estimated value and the covariance matrix of estimated error are calculated by (17), (18) and (19) where $\mathrm{K}_{\mathrm{k+1}}$ is the Kalman gain in the (k+1)th image frame.

$$\mathrm{K}_{k+1} = \mathrm{P}_{k+1|k} H_{k+1}^T \left(H_{k+1} \mathrm{P}_{k+1|k} H_{k+1}^T + R_{k+1} \right)^{-1} \tag{17}$$

$$\hat{\mathrm{x}}_{k+1|k+1} = \hat{\mathrm{x}}_{k+1|k} + \mathrm{K}_{k+1} \left[\mathrm{z}_{k+1} - H_{k+1} \hat{\mathrm{x}}_{k+1|k} \right] \tag{18}$$

$$\mathrm{P}_{k+1|k+1} = \left[I - \mathrm{K}_{k+1} H_{k+1} \right] \mathrm{P}_{k+1|k} \tag{19}$$

During fingertip tracking, we first substitute the optimal estimated value $\hat{X}_{k|k}$ in the kth image frame into (15) to get the optimal predicted value $\hat{X}_{k+1|k}$ in the (k+1)th image frame and obtain the predicted locations in the (k+1)th image frame. Second, we use data association to associate the trajectories of fingertips with the detected locations of fingertips in the $(k+1)^{\text{th}}$ image frame (see Section 4.2). Third, we substitute the detected locations of fingertips in the $(k+1)^{\text{th}}$ image frame into (18) to get the optimal estimated value in the $(k+1)^{\text{th}}$ image frame.

4.2 Fingertip Association

The nearest neighbor method is used for data association with Euclidean distance as the metric. The Euclidean distance between the predicted location and the detected location in the $(k+1)^{\text{th}}$ image frame is calculated by:

$$d = \sqrt{(x - x')^2 + (y - y')^2} \tag{20}$$

The detected location of the j^{th} fingertip in the $(k+1)^{\text{th}}$ image frame is (x_j, y_j) and the predicted locations of fingertips in the $(k+1)^{\text{th}}$ image frame are (x'_1, y'_1), $(x'_2, y'_2)\ldots(x'_n, y'_n)$. The Euclidean distances between the detected location of the j^{th} fingertip and all predicted locations of fingertips are calculated according to (20). If the nearest neighbor of the j^{th} detected fingertip is the i^{th} predicted fingertip and their distance is less than a threshold value, the i^{th} fingertip in the k^{th} image frame is matched to the j^{th} fingertip in the $(k+1)^{\text{th}}$ image frame. Consequently, the j^{th} fingertip in the $(k+1)^{\text{th}}$ image frame is associated with the i^{th} fingertip in the k^{th} image frame to form a trajectory. In this way, we can realize the tracking of multiple fingertips.

Note that new fingertips may appear and existing fingertips may disappear during the process of tracking. If a detected fingertip in the $(k+1)^{\text{th}}$ image frame is not close enough to any predicted fingertips, it is considered as a new fingertip. If the predicted location of a tracked fingertip in the $(k+1)^{\text{th}}$ image frame is not close to any detected fingertip, this fingertip is considered as being disappeared.

In summary, compared with the traditional Kalman filter, the improved Kalman filter eliminates the velocity and acceleration of fingertips as well as the time interval between image frames. In fact, it only uses the locations of fingertips in the previous three image frames to predict the locations of fingertips in the current image frame, which reduces the computational cost and avoids the prediction error caused by the change of time interval between image frames. The proposed method considers the influence of acceleration, which makes the predicted locations of fingertips more accurate and avoids the failure of tracking caused by the sudden change of movement.

5 Experiment and Analysis

Experiments were conducted using a desktop computer with Intel Core i5-2400 at 3.10 GHz CPU and a USB optical camera with 640×480 resolution in the normal lighting environment. The proposed methods were implemented using Visual C++ and OpenCV 2.4.3.

Efficiency. The tracking program achieved ~23 FPS with no finger extended and ~18 FPS when all fingers were extended. The average frame rate of the proposed method was ~20 FPS and the real-time constraint was satisfied reasonably well.

Accuracy. Table 1 shows the accuracy of fingertip detection and tracking with different fingertip numbers. Note that the accuracy of tracking was higher than the accuracy of detection. The reason is that even when there were some interfering points, they were likely to be far away from fingertips and might not be incorrectly associated with fingertips. The main reason of fingertip tracking failure is due to the false locations of fingertips caused by the environmental factors (e.g., illumination change), resulting in wrong fingertip association.

Table 1. Accuracy of Proposed Tracking Method

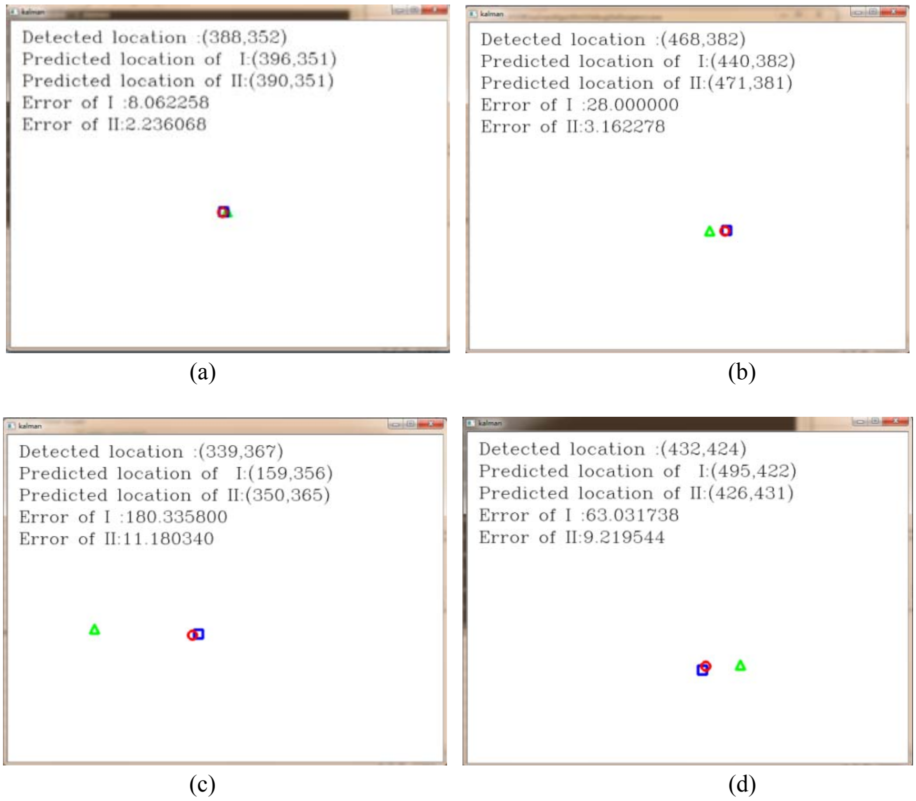
Number of fingertips	**Number of frames**	**Number of correctly detected**	**Number of correctly tracked**	**Accuracy rate of detection**	**Accuracy rate of tracking**
1	200	199	200	99.5%	100%
2	200	198	200	99.0%	100%
3	200	189	200	94.5%	100%
4	200	192	196	96.0%	98.0%
5	200	188	200	94.0%	100%
Total	1000	966	996	96.6%	99.6%

Number of correctly detected is the number of image frames that have only one detected fingertip for each finger and no interfering points;
Number of correctly tracked is the number of image frames that achieve continuous and accurate tracking for each fingertip.

In order to further demonstrate the effectiveness of the improved Kalman filter, a set of illustrative experiments was conducted using the mouse cursor instead of the fingertip. Fig. 4 shows the predication results under different conditions where 'I' represents the traditional Kalman filter and 'II' represents the improved Kalman filter. The red circle, green triangle and blue square are the detected location, the location predicted by the traditional Kalman filter and the location predicted by the improved Kalman filter, respectively.

Intuitively, the improved Kalman filter produced more accurate prediction results and its advantage became more distinct when there was velocity change or direction change. Table 2 shows the distances between the detected locations and the predicted locations using the traditional Kalman filter and the improved Kalman filter in different conditions.

Finally, Fig. 5 shows a typical example of the continuous tracking of five fingertips within around 100 frames in a practical environment. The five trajectories are marked in red and it is clear that although the motions of fingertips were highly nonlinear, the constructed trajectories were smooth and complete, confirming the effectiveness of the proposed tracking method.



```
Detected location :(388,352)
Predicted location of  I:(396,351)
Predicted location of II:(390,351)
Error of I :8.062258
Error of II:2.236068
```

(a)

```
Detected location :(468,382)
Predicted location of  I:(440,382)
Predicted location of II:(471,381)
Error of I :28.000000
Error of II:3.162278
```

(b)

```
Detected location :(339,367)
Predicted location of  I:(159,356)
Predicted location of II:(350,365)
Error of I :180.335800
Error of II:11.180340
```

(c)

```
Detected location :(432,424)
Predicted location of  I:(495,422)
Predicted location of II:(426,431)
Error of I :63.031738
Error of II:9.219544
```

(d)

Fig. 4. Comparison of accuracy between the traditional Kalman filter and the improved Kalman filter under different conditions: (a) Low velocity; (b) High velocity; (c) Velocity change; (d) Direction change

Table 2. The Distances between the Detected Locations and the Predicted Locations

	Low velocity	**High velocity**	**Velocity change**	**Direction change**
The traditional Kalman filter	**8.1**	**28.0**	**180.3**	**63.0**
The improved Kalman filter	**2.2**	**3.2**	**11.2**	**9.2**

The unit of distance is pixel.

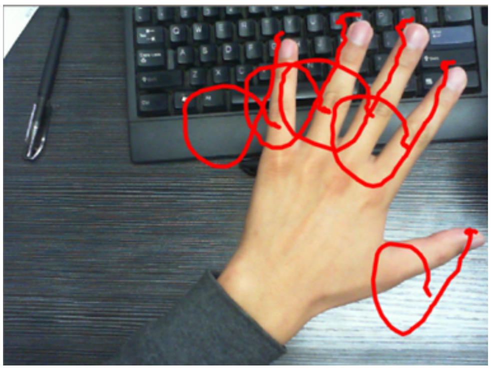


Fig. 5. An illustration of the continuous tracking of five fingertips

6 Conclusions

This paper proposed an effective fingertip tracking approach using a single entry-level USB camera and bare hands. It combines the location, velocity and acceleration information of fingertips and employs the improved Kalman filter to predict the locations of fingertips in the current image frame based on the locations of fingertips in the previous three image frames. Compared with the traditional Kalman filter, it does not need to calculate the velocity of fingertips and the time interval between image frames, which reduces the computational cost and avoids the prediction error caused by the change of time interval between frames. It also considers the influence of acceleration, which makes the predicted locations of fingertips more accurate and avoids the failure of fingertip tracking caused by the sudden change of movement effectively. Furthermore, this paper proposed a method for removing interfering points based on distance transform.

Experimental results show that the proposed method can achieve robust fingertip tracking in a real-time manner and properly handle situations with the appearance of new fingertips as well as the disappearance of existing fingertips. A potential direction for future work is to further improve the robustness of tracking against complex background using machine learning techniques and apply the proposed techniques to interesting real-world applications.

References

1. Wang, L., Hu, W., Tan, T.: Recent developments in human motion analysis. Pattern Recognition 36, 585–601 (2003)
2. Oka, K., Sato, Y., Koike, H.: Real-time fingertip tracking and gesture recognition. Computer Graphics and Applications 22, 64–71 (2002)
3. Xie, Q., Liang, G., Tang, C., Wu, X.: A fast and robust fingertips tracking algorithm for vision-based multi-touch interaction. In: 10th IEEE International Conference on Control and Automation, pp. 1346–1351 (2013)

4. Nakamura, T., Takahashi, S., Tanaka, J.: Double-crossing: A new interaction technique for hand gesture interfaces. In: 8th Asia-Pacific Conference on Computer-Human Interaction, LNCS, vol. 5068, pp. 292–300. Springer, Heidelberg (2008)
5. Jaward, M., Mihaylova, L., Canagarajah, N., Bull, D.: A data association algorithm for multiple object tracking in video sequences. In: the IEE Seminar on Target Tracking: Algorithms and Applications, pp. 129–136 (2006)
6. Letessier, J., Bérard, F.: Visual tracking of bare fingers for interactive surfaces. In: 17th Annual ACM Symposium on User Interface Software and Technology, pp. 119–122 (2004)
7. Wang, X. Y., Zhang, X. W., Dai, G. Z.: An approach to tracking deformable hand gesture for real-time interaction. Journal of Software 18, 2423–2433 (2007)
8. Zarit, B. D., Super, B. J., Quek, F. K.: Comparison of five color models in skin pixel classification. In: International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pp. 58–63 (1999)
9. An, J. H., Hong, K. S.: Finger gesture-based mobile user interface using a rear-facing camera. In: 2011 IEEE International Conference on Consumer Electronics, pp. 303–304 (2011)
10. Gasparini, F., Schettini, R.: Skin segmentation using multiple thresholding. In: Internet Imaging VII, SPIE 6061, pp. 60610F (2006)
11. Chai, D., Ngan, K. N.: Face segmentation using skin-color map in videophone applications. IEEE Trans. on Circuits and Systems for Video Technology 9, 551–564 (1999)
12. Lee, T., Hollerer, T.: Handy AR: Markerless inspection of augmented reality objects using fingertip tracking. In: 11th IEEE International Symposium on Wearable Computers, pp. 83–90 (2007)
13. Lee, B., Chun, J.: Manipulation of virtual objects in marker-less AR system by fingertip tracking and hand gesture recognition. In: 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 1110–1115 (2009)
14. Pan, Z., Li, Y., Zhang, M., Sun, C., Guo, K., Tang, X., Zhou, S. Z.: A real-time multi-cue hand tracking algorithm based on computer vision. In: 2010 IEEE Virtual Reality Conference, pp. 219–222 (2010)
15. Liao, Y., Zhou, Y., Zhou, H., Liang, Z.: Fingertips detection algorithm based on skin colour filtering and distance transformation. In: 12th International Conference on Quality Software, pp. 276–281 (2012)
16. Han, C. Z., Zhu, H., Duan, Z. S.: Multi-source information fusion, 2nd ed.. Press of Tsinghua University, Beijing (2010)
17. Kalman, R. E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82, 35–45 (1960)
18. Schneider, N., Gavrila, D. M.: Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study. In: 35th German Conference on Pattern Recognition, LNCS, vol. 8142, pp. 174–183. Springer, Heidelberg (2013)