Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)

A Hand Direction Vector Extraction Method Based on K Vector and K-medoids Clustering Algorithm

Duan Hong*
Software School of Xiamen University
Xiamen city,China
*corresponding author: hduan@xmu.edu.cn

Cai Kunyu
Software School of Xiamen University
Xiamen city,China
e-mail: 727452830@qq.com

***Abstract*—This paper proposed a method of computing the hand direction vector. At first, using K-vector method, compute the K-slope of all the points on the contour to get a series of peak point sets; Then, cluster the sets which have been obtained through K-medoids algorithm; At last, detect the pointing direction of each fingertip from the clustering information, and get the hand's direction by it. The experiment shows that the method achieves the precise positioning of hand direction in palm contours.**

Keywords-Hand Direction Vector, K-vector Algorithm, K-medoids Clustering Algorithm

I. INTRODUCTION

To extract the hand's direction vector from the palm contour is the key of recognition technology application [1]. Shahzad Mallik [2], W Andrew [3] and others use K vector to do the peak detection of probable fingertips in the hand-shape contour image, and then obtain the hand direction vector. K vector, mentioned by the articles above, is not universal after experimental verification in this paper. This paper discusses how to get K value by experiment.

On the polymerization treatment of the peak point, K-means algorithm [4-7] is a common clustering algorithm. But it's highly vulnerable to be effected by outliers, since using mean as center. Through experiment and discussion, this paper finds that, K-medoids method has the advantage of being not sensitive to outliers, for it using medoids instead of mean as center. It's suitable to apply in the clustering algorithms of fingertips and webbed points. When distinguishing between fingertips and webbed points, we propose a method that to compute the vector cross product. Finally, we finish the detection of fingertip direction by clustering center.

II. K-VECTOR BASED PEAK POINT DETECTION

In the outline of a hand, fingertips and webbed points are usually ones with lager curvature. We call them peak points [1-3]. It uses K-vector method to compute to obtain the peak points. The specific process is as follows:

1) At first, we compute the precursor and the No. K subsequent points of point P(K). Let P(i) = (xi, yi) as the No. i on the contour C, Nc as the total points of the contour, and point PBK(i) which is on the precursor No. k point of P(i) on the contour C as:

$$P_{BK}(i) = \begin{cases} P(i-K), & i-K>0 \\ P(N_C-K+i), & i-K<0 \end{cases} \quad (1)$$

Let point PAK(i) which is on the subsequent No. k point of P(i) as:

$$P_{AK}(i) = \begin{cases} P(i+K), & i+K<N_C \\ P(i+K-N_C), & i+K>N_C \end{cases} \quad (2)$$

2) Then, we compute the cosine cosθ for constant K, one point P on the contour C, vector $\overrightarrow{P_{BK}P}$ and vector $\overrightarrow{PP_{AK}}$. The calculation uses the cosine equation:

$$\cos\theta = \frac{\overrightarrow{P_{BK}P}\cdot\overrightarrow{PP_{AK}}}{|\overrightarrow{P_{BK}P}||\overrightarrow{PP_{AK}}|} \quad (3)$$

3) In order to improve the stability of detection, we take the whole natural numbers of K within a certain range (Kmin < K < Kmax), get the average value after computing in turn. So we can get the K-cosine value KCV(P) of point P. The computational equation of KCV(P) put out by equation (1), (2), and (3) is as:

$$KCV(P) = \frac{1}{Kmax-Kmin+1}\sum_{K_{max}}^{K_{min}}\frac{\overrightarrow{P_{BK}P}\cdot\overrightarrow{PP_{AK}}}{|\overrightarrow{P_{BK}P}||\overrightarrow{PP_{AK}}|} \quad (4)$$

4) After computing the KCV value of each point on contour C, according to the peak point with large curvature, we get the thresholdθT (in this paper, we set -0.7 as the threshold), screen out all the points which are less thanθT , so as to get the peak point sets Peeks = { P(x,y) | KSV(P) < -0.7 } on contour C.

III. THE PEAK POINT CLUSTERING BASED ON K-MEDOIDS ALGORITHM

After getting the peak point set Peeks, we just need to cluster the point sets to obtain the point sets of each fingertip and webbed point. We use K-method clustering algorithm [4-7] to cluster the peak point set Peeks, get a class represented every fingertip and webbed point. The algorithm process is as follows:

1) Arbitrarily select K peak point objects from n peak points as medoids (P1, P2, …Pi…Pk).

2) According to the principle of similar to medoids, assign the rest peak point objects to every class. In this paper, we define the dissimilarity of two points Dis(Pi, Pj) as the absolute value of the index distance of two points on the contour. It's as:

Published by Atlantis Press, Paris, France.
© the authors

$$Dis\left(P_i,P_j\right) = |j - i| \quad (5)$$

3) In each class, sequentially select one object Pr in the class, compute the consumption E(Pr). This paper defines the medoids dissimilarity S(Pi, CLT) of point Pi and a cluster CLT as the dissimilarity of point Pi and the medoids point of the cluster, and define the consumption E(CLT, Pi) of point Pi in a cluster CLT as the sum of point Pi and the dissimilarity of other points:

$$E\left(CLT,P_i\right) = \sum_{j=CLT[0]}^{n} Dis\left(P_i,P_j\right); \quad (6)$$

CLT[0] is the index of the first point on the contour C of cluster CLT, n is the number of the class members in CLT.

4) Choose Pr with the smallest E as the medoids of this cluster.

5) Repeat step 2 until the K medoids are steady.

IV. Fingertip Direction Detection Based on Clustering Center

Before computing the hand direction, we need the cluster CLT of every probable fingertip to get their direction. The steps are as follows:

1) Let the cluster of No.i probable fingertip as CLT_i . Assume its center as $Medoids_i = P$. Compute the vector sum of vector $\overrightarrow{P_{BK}P}$ and $\overrightarrow{P_{AK}P}$ when K is a constant:

$$\overrightarrow{P_{Ksum}} = \overrightarrow{P_{BK}P} + \overrightarrow{P_{AK}P} \quad (7)$$

2) Sequentially let K as the natural number in the interval [Kmin, Kmax], Kmin > 0. Compute the sum and use the average number as the mean of fingertip vector:

$$\overrightarrow{P_{avg}} = \frac{1}{Kmax - Kmin + 1} \sum_{Kmax}^{Kmin} \overrightarrow{P_{Ksum}} \quad (8)$$

3) Normalize the vectors to get the fingertip vector . It's shown as Figure 1:

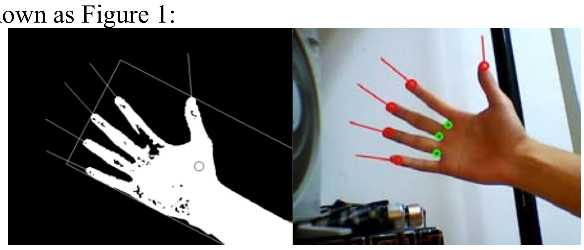

Figure 1. Schematic diagram of fingertips vector

V. The Hand Direction Getten Based on Fingertip Direction

Hand direction refers to the direction which points the arm to the palm, shown as Figure 1. In this paper, we will make use of hand direction to screen suspected fingertips and exclude mistaken points. The hand direction can also be used as the judgement basis of the direction of movement of the hand. The main purpose of this section is to get the hand direction $\overrightarrow{Vh}$ by existing fingertips.

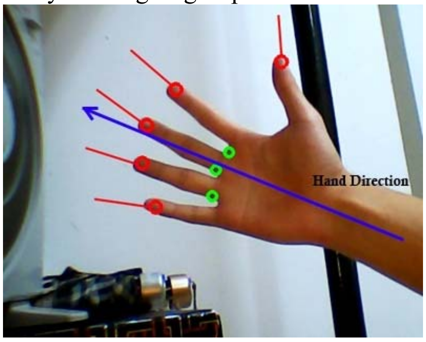

Figure 2. The vectors of hand direction

After getting vectors of all the suspected fingertips, we can make the following conclusions by observing and analysing Figure 1:

When the hand has one or more fingertips, a certain fingertip vector among them can be used as its hand direction. As shown in Figure 1, we can use index finger, middle finger or ring finger as the direction vector of the hand.

When we have more than 3 fingertips, it's usually the middle finger or the index finger to be used as the direction vector of the hand. Then further compute the sum of the angle difference between fingertip vectors to get the data in Table 1. After analysing it, we find that the fingertip vector $\overrightarrow{Vi}$, which can be used as hand direction, has following property: The sum of angle difference between this fingertip vector and others is the lest in all the fingertip vectors.

TABLE I. The Sum of Angle Difference Between Fingertips (Degree Measure)

Fingertip name	4 fingertips	5 fingertips
Thumb	-	157°
Index finger	40°	95°
Middle finger	49°	80°
Ring finger	40°	91°
Little finger	45°	116°

It means that the fingertip vector of the hand direction satisfies the following conditions:

$$\overrightarrow{Vh} = Min(\theta_{sum}(I)); \quad (9)$$

$$\theta_{sum}(I) = \sum_j arccos\left(cos\overrightarrow{V_i}\overrightarrow{V_j}\right); \quad (10)$$

$cos\overrightarrow{V_i}\overrightarrow{V_j}$ is the cosine of vector $\overrightarrow{V_i}$ and $\overrightarrow{V_j}$.

As there are several mistaken points in the suspected fingertips we detect, it may not accurately find the fingertip vector which can be used as the hand direction in the case of little fingertips. So we use this method in the case of more fingertips, so as to get the hand direction accurately.

In this paper, we make the hand with 5 fingers spread as the starting state in the video. So there are many fingertips in the initial state. It can judge the hand direction by fingertip vector steadily. When it's in the normal video frame rate, the change of the hand between two frames is very few, so in the subsequent video frames, we can detect the hand direction by last one. Compare them and choose one that is similar to the last frame as the hand direction.

The specific algorithm process is as follows:

1) When in the initial state, for the hand with 5 fingers spread, find a fingertip vector as the direction vector $\vec{Vh}$ of this hand according to formula 9 and 10.

2) When in the condition of having got the direction vector in this frame:

- Compute the number of fingertips in this frame, determine whether it's smaller or equal to three.
- When the angle between the vector and one in the last frame is more than 90°, exclude the vector.
- Compute as formula 9 and 10, choose what is nearer to the last frame as the direction vector $\vec{Vh}$ of the hand.

Through the process above, we can finally obtain a steady direction vector $\vec{Vh}$. The effect is as Figure 3. The blue line is the fingertip vector which is as direction vector of the hand. In the 2 pictures of Figure 3, the fingertip vector of middle finger represents the direction vector.

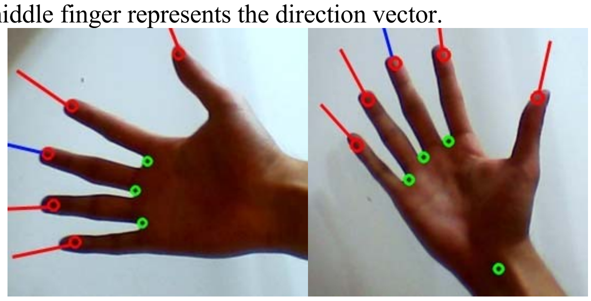

Figure 3. The effect of hand direction vectors

VI. THE EXPERIMENT AND CONCLUSION

In the experiment, we prepare five groups of sample figures, in which the included angle between hand and the level of the left direction are separately 0, 45, 90, 135 and 180 degree. The samples are gotten from the video screenshots whose resolution of video equipment is 640*480, and the frame rate is 45 frames/sec. The samples are shown as Figure 4.

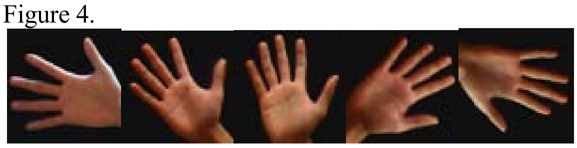

Figure 4. Five groups of samples(It's 1 to 5 from left to right)

Every group of samples has 100 pictures. We extract the hand vectors from each one, and analyse the fingertips selected by the vectors. The results are as Table 2. The fingertips concentrated in forefinger, middle finger and ring finger. It satisfies the requirements of the subsequent gesture recognition.

TABLE II. THE NUMBER OF FINGERTIPS VECTOR BEEN CHOSEN AS HAND VECTOR IN THE SAMPLE

Fingertip	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Thumb	5	9	0	3	6
Forefinger	21	32	33	23	32
Middle Finger	34	31	35	29	31
Ring Finger	31	26	32	30	21
Little Finger	9	2	10	15	10

This paper proposes a method of calculating the hand direction vector. In this method, we use K-vector method to get a series of peak point sets of palm contour at first; Then, use K-medoids algorithm to cluster the sets which has been obtained before, and get the clustering of every fingertip and webbed point; At last, we finish the detection of fingertip direction by cluster center. After experimental verification, it will have an accurate effect when the video rate is between 30 and 45 frames.

REFERENCES

[1] Kangde Guo, "Vision-based three-dimensional fingertip detection algorithm and application," Zhejiang University, 2010.

[2] Shahzad Malik, "Real-time hand tracking and finger tracking for interaction," CSC2503F Project Report, 2003.

[3] W Andrew, S Mubarak, D Vitoria, "A virtual 3D blackboard: 3D finger tracking using a single camera," IEEE International Conference on Automatic Face and Gesture Recognition, pp.536-543, 2000.

[4] Richard Roiger, Michael Geatz, "Data Mining: A Tutorial-based Primer," Pearson Education, 2003.

[5] Raghu Krishnapuram, Anupam Joshi, Liyu Yi, "A Fuzzy Relative of the k-Medoids Algorithm with Application to Web Document and Snippet Clustering," Snippet Clustering, in Proc. IEEE Intl. Conf. Fuzzy Systems – FUZZIEEE99, 1999.

[6] T.Velmurugan, T.Santhanam, "Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points," Journal of Computer Science, Vol. 6, No. 3, pp.363-368, 2010.

[7] http://baike.baidu.com/view/69222.htm

[8] Kunyu Cai, "Fingertips Recognition and PowerPoint Display Control Based on Computer Vision," Bachelor Thesis of Xiamen University, 2012.

© the authors