A Fingertip and Webbed Point Detection 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*—The paper presents a calculation method to get the probable fingertips and webbed points in outline. First, use K-vectoe to calculate the K-slope of all points in outline and get a series of peak point sets. Then, cluster the peak point sets which have been obstained by K-medoids algorithm and get the location of every fingertips and probable points; At last, distinguish the fingertips and webbed points by vector cross product operation. The experiment shows that this method achieve the precise positionong of the fingertips and finger webbed points.**

Keywords-Fingertip Detection , Webbed point Detection,K-vector Algorithm, K-medoids Clustering Algorithm

I. Introduction

The palm contour fingertips positioning id the key of gesture recognition technology application [1]. Shahzad Malik [2], W Andrew [3] and others use K-vector method to do the peak detection of the suspected fingertip in the hand-shaped contour image. The process only takes vector point multiplication, and uses a smaller constant K to accurately calculate the curvature of each point on contour C. It has the features of low complexity and high accuracy. But the K-vector put forward above is not universal after the experimental certification in this paper. This paper discusses the obtainment of K value by experiment.

On the polymerization treatment for the peak points, K-means algorithm [4-7] is a common clustering algorithm. But as it uses the mean as the center, it is highly vulnerable to be affected by outliners. This paper finds through experiments and discussions that, as K-medoids algorithm [4-6] uses medoids as center instead of the mean, it's not sensitive to outliers. So it's very availiable to fingertips and finger webbed points cluster computing.

At last, we propose that the method of computing vector cross product can easily distinguish fingertips and finger webbed points.

II. The Peak Point Detection Based on K-vector

By observing the palm outline in Figure 1, we can draw the following conclusion: in the outline of a hand, fingertips and finger webbed points are usually the greater curvature points in the outline, we call it peak points [1-3]. To get them, we use K-vector method.

K-vector method need to specify the traversal direction and starting point of the contour. We can know from literature that, the starting point of the contour P(0) = (x0, y0) is the pixel with smallest y-coordinate value. If it has a few pixels with the same y-coordinate value, we make those with smallest x-coordinate as the starting point of the contour. The contour traversal direction is counterclockwise.

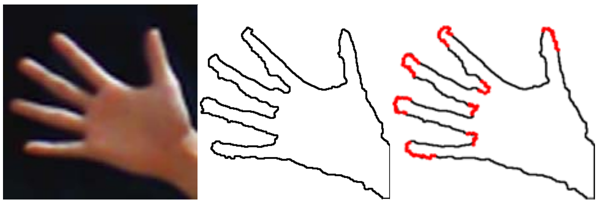

Figure 1. Hand contour map

The specific process to get the peak point sets by K-vector method 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:

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

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

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

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:

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

3) In order to improve the stability of detection, we take the whole natural numbers of K within a certain range ($K_{min} < K < K_{max}$), 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:

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

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

Published by Atlantis Press, Paris, France.

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. As shown in Figure 1.

In the experiment, we use the home video equipment with 640*480 resolution, and intercept 300 sample maps for test. These 300 sample maps are cut out in different distance from hand to cemera. It contains the hand with 5 fingers in the sample maps. We make the hand contour extraction for sample maps, and get 300 hands' full contour maps.

We use K-vector method to pick up the peak point sets of 300 hand contour maps, analys the experimental results, and verify the accuracy of the algorithm and the main conditions of erroneous detection.

By the experimental data, we find that, when the hand is closer to the video, the K value which is between 50-70, is the larger the better; when the distance is far, it's the less the better which is between 30-50. The experimental results are shown as Table 1:

TABLE I. THE TEST ACCURACY IN THE SAMPLES WITH DIFFERENT DISTANCES WHEN USING K-VECTOR TO GET THE K VALUE

K value	A distance about 30 cm	A distance about 50 cm	A distance larger than 60 cm
30-40	85%	92%	92%
40-50	85%	93%	89%
50-60	88%	90%	87%
60-70	92%	87%	85%

As the operating distance is of 50 cm and above in the real environment, we choose the threshold value range between 40-50 to get the better result in actual operation.

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

After getting the peak point set Peeks, we observe the result (Figure 2), and know that the peak points are uniform gathered near the fingertips and finger webbed points. We just need to cluster the point sets to obtain the point sets of each fingertip and webbed point.

In this section, we use K-method clustering algorithm [4-7] to cluster the peak point set Peeks, get a class represented every fingertip and webbed point, and compute every point's coordinate averge as the representative point. The process of K-medoids algorithm 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:

$$\mathrm{Dis}(P_i, P_j) = |j - i| \tag{5}$$

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:

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

Among them, 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.

In the experiment, we use ordinary camera with 640*480 resolution ratio. For the peak point sets got in the case of a given threshold value K in section 2.1, we do the cluster with K-means and K-medoids algorithms to obtain the result of the peak point set. The result is shown as Table 2:

TABLE II. THE CORRECT RATE OF K-MEANS AND K-MEDOIDS ALGORITHMS FOR PEAK POINT SETS

Clustering algorithm	Number of correctly classified	Number of misclassified	Correct rate
K-means Algorithm	2354	135	94.57%
K-medoids Algorithm	2457	32	98.71%

In the test, we find that there are some anomalous peak points outside the point set range we expected. As K-means algorithm uses sample mean as the center, it's easy to be affected by abnormal value and the error detection value, so the sample center is offset and it leads to inaccurate classification. While K-medoids algorithm is insensitive to outliers for using medoids instead of mean as center.

Through K-medoids clustering algorithm, after achieving the cluster in the range of every fingertip and webbed point, we can compute the coordinate mean of all the points in each cluster, to get the position of fingertips and webbed points we need as $PF = (\frac{1}{n}\sum x_i, \frac{1}{n}\sum y_i)$. The result is as Figure 3, the circle position represents determination probable fingertips or webbed position:

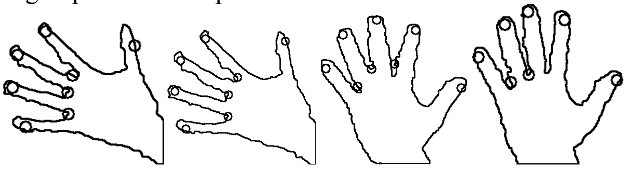

Figure 2. Cluster effect picture

IV. DISTINGUISH THE PROBABLE FINGERTIPS AND WEBBED POINTS BASED ON CROSS PRODUCT OF VECTORS

In the previous section, we get the coordinate position of probable fingertips and webbed points. In this section, we will distinguish them by computing the cross product of vectors.

Definition 1 In a three-dimensional vector space, assume that a and b are two vectors, then their cross product can be strictly defined as follows:

- |c|=|a×b|=|a||b|sin<a,b>
- $c \perp a$， and $c \perp b$
- The direction of c is judged by the right-hand rule.

Vector cross product has a very important character, it's that we can determine the clockwise and counterclockwise relationship of two vectors by the symbol of vector cross product.

- if P×Q > 0, P is in the clockwise direction of Q.
- if P×Q < 0, P is in the counterclockwise direction of Q.
- if P×Q = 0, P and Q are collinear, but may also be in the same or reverse direction.

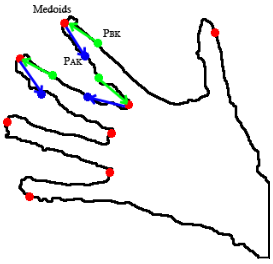

Figure 3. Vector diagram

We take clustering information gotten by K-medoids algorithm. Let point P as the medoids of cluster CLT. Through several experiments, in different distances from hand to camera, compute respectively the vector cross product of fingertip and webbed point $\overline{P_{BK}P}$ and $\overline{P_{AK}P}$, and get the results of Table 3 and Table 4.

TABLE III. TABLE 3 THE VECTOR CROSS PRODUCT SYMBOL OF FINGERTIPS $\overline{P_{BK}P}$ AND $\overline{P_{AK}P}$

Fingertips	**Sample 1 Close distance**	**Sample 2 middle distance**	**Sample 2 long distance**
Thumb	positive number	positive number	positive number
Forefinger	positive number	positive number	positive number
Middle finger	positive number	positive number	positive number
Ring finger	positive number	positive number	positive number
Little finger	positive number	positive number	positive number

TABLE IV. TABLE 4 THE VECTOR CROSS PRODUCT SYMBOL OF WEBBED POINTS $\overline{P_{BK}P}$ AND $\overline{P_{AK}P}$

Webbed points	**Sample 1 Close distance**	**Sample 2 middle distance**	**Sample 2 long distance**
Point 1	negative number	negative number	negative number
Point 2	negative number	negative number	negative number
Point 3	negative number	negative number	negative number

By observing the data of Figure 3 and Table 4, we can find that, $\overline{P_{BK}P}$ and $\overline{P_{AK}P}$ in fingertips and webbed points have different interrelation, and the vector cross product have different symbols. Since the traversal direction of the contour C is counterclockwise, we can know:

- Fingertip $\overline{P_{BK}P}$ is in the clockwise direction of $\overline{P_{AK}P}$, it conforms that $\overline{P_{BK}P} \times \overline{PP_{AK}} > 0$
- Fingertip $\overline{P_{BK}P}$ is in the counterclockwise direction of $\overline{P_{AK}P}$, it conforms that $\overline{P_{BK}P} \times \overline{PP_{AK}} < 0$

So we can distinguish fingertip and webbed point by determining the symbol of vector cross product. The result is shown as Figure 5 in which the red circle represents the detected fingertip and the green as webbed point:

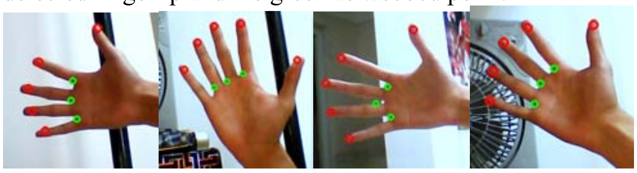

Figure 4. The Effect picture of fingertips and webbed points based on vector cross product

V. CONCLUSION

This article describes how to compute to get the fingertips and webbed points of the contour, according to the existing contour information.

At first, we use K-vector method to compute the K-slope of every point on the contour, and obtain a series of peak point set; Then, use K-medoids algorithm to cluster the existing peak point sets, and obtain the cluster of every fingertip and webbed point; At last, we distinguish them by computing vector cross product.

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