Fingerprint Recognition Techniques

Pinnap pranathi[1], G.Rupa kinnera[2] & Dr. G. Vishnu Murthy[3]

[1, 2] Mtech 2nd year: Dept of CSE. CVSR College of Engineering, Hyderabad, India

[3] Associate Professor and HOD: Dept of CSE. CVSR college of Engineering, Hyderabad, India

Abstract:

Convex hull is a popular structure in computational geometry. It's a useful tool in constructing structures like Voronoi diagrams, and we use this structure for fingerprint recognition. This paper utilizes the Graham's scan algorithm for computing convex hull in a specified space which is tested for its reliability, performance, uniqueness and stability. Here we analyze and compare the time, space complexities with the K-NN clustering for better performance. The Graham's scan algorithm produces an accurate complexity of O(n log n) despite the size of the hull. The key concept is to pre process the input and uses the order to help decide which points to pick. This work also contributes to the pattern recognition and cluster orientation.

Keywords

Voronoi diagrams, Graham's scan algorithm, K-NN clustering.

1. Introduction

Biometrics is a robust verification procedure used in various sectors such as enterprise, government, finance and banking. A biometric system substitutes the need of physical ID, passwords [1] and serves as an indisputable proof for a person's identity. Biological features such as fingerprint, face, voice etc are used by the biometric security technologies to avoid unauthorized access. As biometric authentication systems have many advantages over traditional security systems, still few drawbacks exist. Such as biometric data are consistently good in quality but noisy by nature and when a person's identity is lost, it can't be replaced.

Here we consider fingerprint as the biological feature for authentication. The fingerprint consists of minutia, points of interest having ridge bifurcations and ridge endings. Each minutiae is matched by its position (x,y,θ) where x and y are coordinates of minutiae points, θ is the orientation angle. There are different categories for biometric template protection such as feature transformation, biometric cryptosystem and hybrid system.

As of now various methods are proposed for protecting the fingerprint template of an individual. In alignment based free methods local and global features of minutiae are considered for transformation [2-3]. Statistical methods adopt the extraction of various mathematical characteristics to classify the patterns [4]. However some hybrid approaches are looked up which combine the structural and statistical methods [5]. The Graph Based K-Nearest Neighbor Minutiae Clustering for Fingerprint Recognition for pattern matching and clustering of graph databases [6]. This paper [7] describes the matched scores obtained from the fusion using weighted sum rule and T-operators.

In [8] proposed a similar application based on fingerprints and used a technique called a fuzzy vault, which was first introduced by Juels and Sudan. In [9] describes an algorithm to match the fingerprint quadruplets that reduces time and space complexities. We use the well known algorithm which stands better performance than the simple convex hull is Graham's scan. This algorithm was named after Ronald Graham, who published the original algorithm in 1972[10]. The Graham's scan works on the convex space rather on concave space [11]. The point triangulation method uses a set of points in the data set to form series of triangle queue or triangle tree [12].

2. Proposed Method

There are two methods proposed in this paper for recognition of fingerprints. They are as follows:

(1) Graham's scan fingerprint recognition.
(2) Fingerprint recognition by K-NN clustering of the minutiae.

METHOD 1: GRAHAM'S SCAN FINGERPRINT RECOGNITION

A Convex hull is defined as a subset S of a plane is called convex if and only if, for any pair of points p,q ∈ S, is completely contained in S [10]. The convex hull CH (S) of a set S is the smallest convex set that contains S as shown in fig 1. The black dots represent the vertices of the convex hull and the red dots are interior points. For computing convex hull Graham's scan algorithm is used. Graham's scan is a method of finding the convex hull of a finite set of points in the plane with time complexity O(*n* log *n*) [9] and then applies a linear-time scanning algorithm to build the hull.

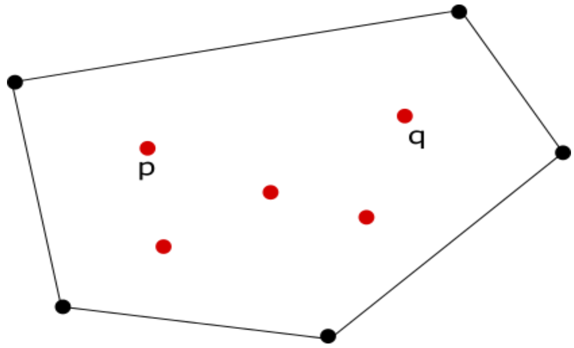

Fig 1 Convex Hull for a set of points

We implement Graham's scan by finding the leftmost point *u* and then sort the points in counterclockwise order around *u*. To compare two points *p* and *q*, we check whether the triple *u*, *p*, *q* is oriented clockwise or counterclockwise. Once the points are sorted, we join them in counterclockwise order, starting and ending at *u*. The result is a simple polygon with n vertices.

To change this polygon into the convex hull, we apply the following 'three-penny algorithm' which follows the two conditions:

1. If p, q, r are in counterclockwise order as shown in fig 2, move the back penny forward to the successor of r. This condition applies for n-2 times.
2. If p, q, r are in clockwise order, remove q from the polygon, add the edge pr, and move the middle penny backward to the predecessor of p. This condition applies for n-h times.

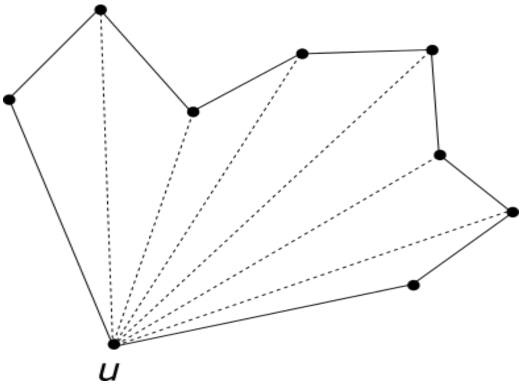

Fig 2 Polygon formed in the sorting stage of Graham's scan

Therefore each counterclockwise test takes constant time, the scanning phase takes O (n) time altogether.

A triangulation method is based on constructing convex layers by Graham's scan. It allows to develop an algorithm with the optimal complexity of O (N log N) [12]. After performing the Graham scan algorithm fig 3 is obtained.

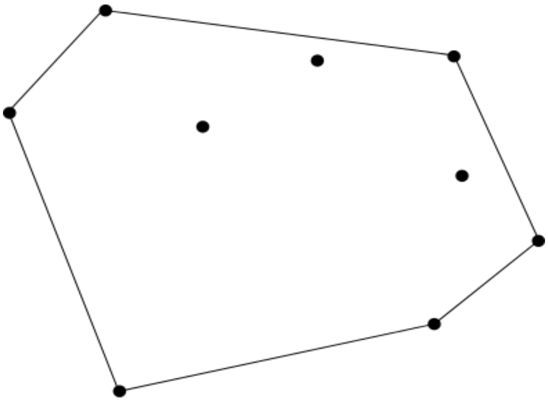

Fig 3 Polygon after implementing Graham's scan

For constructing the Graham's scan algorithm we consider a simple convex hull constructing algorithm.

Algorithm for CONVEXHULL (M)

Input: Consider a set M of minutiae points.
Output: A list containing the vertices of CH (M) in clockwise order.
Begin
Sort the minutiae points in x coordinate with series mp_1 to mp_n.
Point's mp_1 and mp_2 are in a set S_U, with mp_1 as the first point. // S_U is upper hull
for *c* ← 3 to n

do Append mp_i to S_U.

while S_U > 2 points && last 3 points take a right turn.
do Delete the middle of the last three points from S_U.
Put the points mp_n and mp_{n-1} in a list S_L, with mp_n as the first point. // S_L is lower hull
for *c* ← n−2 down to 1
do Append mp_i to S_L.
while S_L> 2 points && last 3 points don't make a right turn.
do Delete the middle of the last 3 points from S_L.
Delete the first and the last point from S_L to avoid matching of points where the upper and lower hull meets.
Append S_L to S_U
return S
end

METHOD 2: FINGERPRINT RECOGNITION BY K-NN CLUSTERING OF THE MINUTIAE

K-NN (K- Nearest Neighbors) clustering method groups the minutiae for K nearest neighbor in the fingerprint image. We consider the minutiae points $M_i = (x_i, y_i, \theta_i)^N$, where (x_i, y_i) are x and y coordinates of a minutiae point, and θ is the orientation angle. The Euclidean distance between minutiae points is calculated as, for Cartesian coordinates, if $\mathbf{x} = (x_1, x_2,..., x_n)$ and $y = (y_1, y_2,..., y_n)$ are two points in Euclidean *n*-space, then the distance (d) is given by,

$$d(x,y)= d(y,x)=\sqrt{(y_1-x_1)^2+(y_2-x_2)^2+\cdots+(y_n-x_n)^2}$$

$$=\sqrt{\sum_{i=1}^{n}(yi-xi)^2}$$

The Euclidean distance vector is given by:

$$|y-x| = \sqrt{|x|}$$

The K-NN clustering proves to be ideal as lower execution time and good clustering accuracy. The K-nearest neighbor algorithm (K-NN) is used for clustering the fingerprint graph nodes based on Euclidean distance between them. Here a node is classified by majority votes of its neighbors. The nodes are assigned to a set that is closest among its K nearest neighbors, where K is a small positive integer. If K = 1, then the node is assigned to the set of its nearest neighbor.

1. Each node within the feature set has a set label, *Set* = $s_1 \ldots s_n$

2. The K-nearest neighbors are calculated by distance matrix where K is the number of neighbors.

3. The K-closest nodes are analyzed to determine which set label is frequent.

4. The frequent set is assigned to the node being analyzed.

3. Implementation

The fingerprint identification is performed on collection of fingerprint images of various individuals. In fingerprint recognition we implemented following algorithms first method Graham's scan fingerprint recognition and second method fingerprint recognition by K-NN clustering of the minutiae. We observe that Graham's scan algorithm uses sorting time complexity O(n log n) and scanning takes O(n). Overall time complexity is O(n log n) as sorting dominates the time to compute the hull. The space complexity of the Graham's scan algorithm is given by O(n). The time complexity of K- Nearest neighbors is given as $O(dN)$ where N is the group of S and d is the dimensionality of M. Here S is a set of points in space M. We measure the execution time of the algorithm for each input set and calculate an average time. We repeat the test for different sizes of an input set. The results of the test are depicted in fig 4 and fig 5 which compares both the proposed methods.

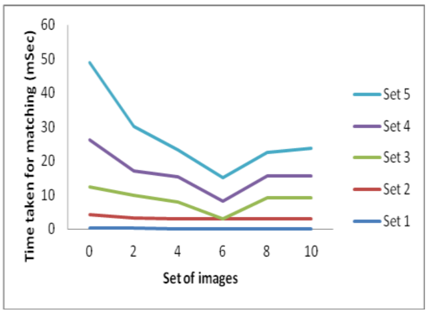

Fig 4 Matching time for a set of images (each hull) in convex hull by Graham's scan

Fig 5 Comparing both methods Graham scan's (method 1) and K-NN clustering (method 2)

4. Conclusion

This paper analysis the complexities and limitations of convex hull computation by Graham's scan algorithm with the K-NN clustering. Both the methods have their unique characteristics for obtaining the minutiae points and constructing each layer of grouping. As obtained from the above graph K-NN clustering method dominates the Graham scan's for matching the fingerprint images, although the method 1 achieves good complexity result. Both methods achieve the minimum requirement of space and time complexities with accuracy. The proposed algorithm also limits the usage of storage space and can be linked to data structures for better stability.

5. References

1. Ratha N., Chikkerur S., Connell J., et al., 'Generating cancelable fingerprint templates', IEEE Trans. Pattern Anal. Mach. Intell., 2007.
2. Ahmad T., Hu J., Wang S., 'Pair-polar coordinate-based cancelable fingerprint templates', Pattern Recognition , 2011.
3. Prasad M.V.N.K., Santhosh Kumar, C., 'Fingerprint template protection using multiline neighboring relation', Expert Syst. Appl., 2014.
4. A.K. Jain, S. Prabhakar, and L. Hong, 'A Multichannel Approach to Fingerprint Classification', IEEE Trans.
5. A. Serrau, G.L. Marcialis, H. Bunke, and F. Roli, 'An experimental comparison of fingerprint classification methods using graphs', Springer.
6. Vaishali Pawar, Mukesh Zaveri, 'Graph Based K-Nearest Neighbor Minutiae Clustering for Fingerprint Recognition', 10th International Conference on Natural Computation 2014.
7. Mulagala Sandhya, Munaga V. N. K. Prasad, 'Multi-algorithmic cancelable fingerprint template generation based on weighted sum rule and T-operators', Springer 2016.
8. Juels and M. Sudan, 'A fuzzy vault scheme', Proceedings of IEEE International Symposium on Information Theory, 2002.
9. A. Tirupathi Rao, N. Pattabhi Ramaiah,V Raghavendra Reddy, C. Krishna Mohan, 'Nearest Neighbor Minutia Quadruplets based', IEEE 2015.
10. Graham R.L., 'An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set', Information Processing Letters, 1972.
11. De Berg M., O. Cheong, M. van Kreveld, M. Overmars, 'Computational Geometry: Algorithms and Applications' (Springer, 2008).
12. Vasyl Tereshchenko, Yaroslav Tereshchenko, Dmytro Kotsur, 'Point triangulation using Graham's scan', INTECH 2015