LEARNING EFFICIENT CODES FOR 3D FACE RECOGNITION

Cheng Zhong, Zhenan Sun and Tieniu Tan

National Laboratory of Pattern Recognition
Institute of Automation Chinese Academy of Sciences

ABSTRACT

Face representation based on the Visual Codebook becomes popular because of its excellent recognition performance, in which the critical problem is how to learn the most efficient codes to represent the facial characteristics. In this paper, we introduce the Quadtree clustering algorithm to learn the facial-codes to boost 3D face recognition performance. The merits of Quadtree clustering come from: (1) It is robust to data noises; (2) It can adaptively assign clustering centers according to the density of data distribution. We make a comparison between Quadtree and some widely used clustering methods, such as G-means, K-means, Normalized-cut and Mean-shift. Experimental results show that using the facial-codes learned by Quadtree clustering gives the best performance for 3D face recognition.

***Index Terms*—** Face recognition, Pattern clustering methods, Image texture analysis, Pattern recognition, Image analysis

1. INTRODUCTION

Recently representations based on loose collections of invariant local texture descriptors extracted from image patches become very popular in texture analysis and object recognition fields. Leung et al. [1] proposed a three-dimensional textons to represent and recognize textures. In their method a vocabulary of prototype tiny surfaces patches was constructed with associated local geometric and photometric properties, which were called as 3D-textons. Then they characterized images of any texture using these clustered 3D-textons. Agarwal et al. [2] proposed hyperfeatures to code different levels of images for object recognition. Their method was to generalize and formalize the above process to higher levels of image coding. And the resulting higher-level features were called as hyperfeatures.

Because of the excellent performance of visual codebook based method in texture analysis and object recognition, it has also been introduced into face recognition area. Xin et al. [3] proposed a novel generative model based on Local Visual Primitives (LVP) for face modeling and classification. In their method, the LVPs were learned by clustering a great number of local facial patches, such as eyes, nose and mouth. Experimental results showed that the learned LVPs are very effective for face reconstruction and recognition. Zhong et al. [4] introduced the Learned Visual Codebook (LVC) method into 3D face recognition. In their method, the visual codes were learned by clustering the Gabor filter response vectors of the 3D faces and the Learned Visual Codebook was constructed based on these learned facial codes. Face recognition was achieved by histogram matching. As shown in Fig.1, the main idea behind the above methods is to construct a comprehensive dense or sparse codebook to represent the images, which can be characterized by quantizing the local texture patches with the learned codes stored in the codebook. Thus, how to learn the effective codes becomes a critical problem in the visual codebook framework. However, most of the methods [1] [3] [4] only adopt K-means clustering for learning.

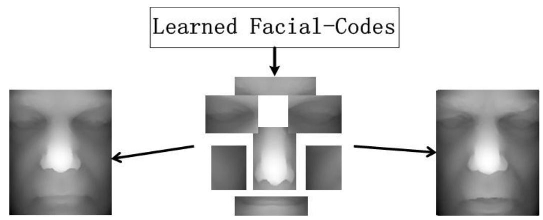


Fig. 1. An example image of the learned facial-codes from 3D faces.

The main contribution of this paper is the introduction of Quadtree clustering, which is to learn the most efficient facial-codes for 3D face recognition. There are already some papers on how to cluster the codes in visual codebook for object recognition [5] [6], however, these papers are based on object recognition, which can be viewed as coarse texture classification. While 3D face recognition can be viewed as fine texture classification, which need more delicate codes for description. We also make detailed comparisons between Quadtree and some commonly used clustering methods, such as G-means [7], K-means [8], Normalized-Cut(Ncut) [9] and Mean-Shift(Mshift) [10].

The remainer of this paper is organized as follows. In Section 2, we introduce the details of Quadtree clustering. We describe our experimental results in Section 3. Finally, the paper is concluded in Section 4.

2. QUADTREE CLUSTERING

How to learn the efficient codes to represent the 3D faces is a critical procedure in the visual codebook based recognition framework. And clustering methods are the widely adopted solutions for this specific problem. Clustering can be viewed as the unsupervised classification of patterns into groups, which has been studied by researchers using many disciplines [11]. The distribution of our 3D facial data is shown in Fig.4(a). From this figure, we can find that there is no suitable models to describe this disordered feature space, in which data density is the only valuable information for clustering. To accurately represent 3D faces using the clustering centers, we first give three rules to evaluate the performance of clustering methods.

Rule1: For data in high-density area

clustering will assign more centers.

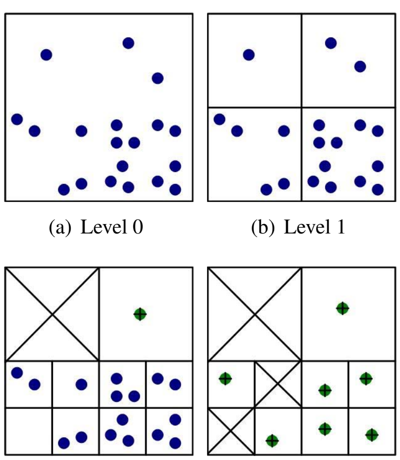
Rule2: For data in low-density area

clustering will assign a few centers.

Rule3: For noisy data

clustering will not assign centers.

Rule1 is to enforce the representing and discriminative ability of the learned centers. Rule2 is to guarantee that the learned centers can cover the entire data space. Rule3 is to guarantee the accuracy of the learned centers.



(a) Level 0 (b) Level 1

(c) Level 2 (d) Level 3

Fig. 2. A toy example of Quadtree clustering.

Quadtree structure has been widely adopted in computer graphics for culling [12]. Motivated by this structure, we give a Quadtree clustering to solve the efficient codes learning problem for the visual codebook based recognition framework. The main procedures of Quadtree clustering can be described as Algorithm.1:

Algorithm 1 Quadtree Clustering

Initialization: two thresholds $thr1$ and $thr2$, an empty stack to store centers, a grid stack to store grid data with total training data as the first element, the termination size of the gird min_grid
while $size(grid\ stack) > 0$ **do**
  Pop the grid data in the top grid stack
  if $NG <= thr1$ **then**
    Discard this grid data
  else if $thr1 < NG <= thr2$ **then**
    Compute the center of this grid and push it into the center stack
  else
    if $SG > min\_grid$ **then**
      Evenly divide this grid into four smaller grids and push them into the grid stack
    else
      Compute the center of this grid and push it into the center stack
    end if
  end if
end while
Pop the centers in centers stack

In Algorithm.1, NG is the number of points in each grid and SG is the size of the grid. $Thr1$ and $thr2$ are critical parameters to control the clustering, which can estimated from the data distribution of training data. In according to the above three rules, if $NG <= thr1$, this is the noisy data area, so we discard the grid data. If $thr1 < NG <= thr2$, this is the low-density data area, so we directly give this gird a center. If $NG > thr2$, this is the high-denity data area, so we divide the grid for further processing. A toy example of our proposed Quadtree clustering is shown in Fig.4(b), in which $thr1 = 1$ and $thr2 = 2$.

The main drawback of Quadtree clustering is that this method is only suitable for the low dimensional data (Although Quadtree is only for 2-dimensional data, it can easily extend to low-dimensional data, such as Octree [12]). However, as Table1 shown (the detail information of the experiment setup is in next section), the recognition performance using the 1 scale and 2 orientation (1S2O) Gabor filters is better than that using the 5 scale and 4 orientation (5S4O) Gabor filters. The main reason for this result is the clustering. The high dimensional data introduces more noises than the low dimensional data into the clustering centers, which will influence the accuracy of the learned centers. Experimental results mean that we only need to carry out the clustering methods in the low dimensional data space for 3D face recognition.

Therefore, Quadtree clustering is suitable for our problem.

Table 1. Comparisons of recognition performances for different dimensions.

Methods	EER	VR
1S2O	3.4%	85.9%
5S4O	5.7%	70.1%

3. EXPERIMENTAL RESULTS

Our proposed Quadtree clustering is evaluated in terms of the representing and discriminative ability of its clustering centers for 3D face recognition based on FRGC2.0 3D Face Database [13].

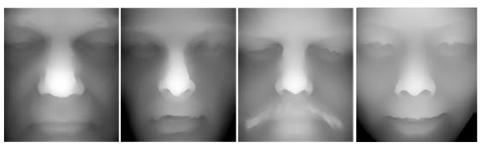


Fig. 3. Some example images in FRGC2.0 3D Face Database.

FRGC2.0 3D Face Database is the most challenging database as far as we know [13], which contains variations of sessions, expressions, illuminations and so on. Some example images from FRGC2.0 3D Face Database are shown in Fig.3. In this experiment we don't follow the rules as FRGC2.0, which use the 943 images in FRGC1.0 as training set and the left 4007 3D face images as testing set. Instead of that, we only choose the first 100 3D faces from this database as training set, and use all of the FRGC2.0 database, 4950 3D face images in total, as our testing set. Here we adopt the Learned Visual Codebook (LVC) framework for 3D face recognition [4]. In clustering procedure of LVC, which is a critical procedure aimed to learn the codes, our proposed Quadtree clustering is compared with some widely clustering methods, G-means [7], K-means [8], Ncut [9] and Mshift [10]. Experimental results are shown in Table.2, EER is the equal error rate and VR is the verification rate when false accept rate is 0.1%.

Table 2. Comparisons of recognition performances for different clusterings.

Methods	EER	VR
Gmeans	3.2%	84.1%
Kmeans	3.4%	85.9%
Ncut	4.4%	77.7%
Mshift	4.0%	76.5%
Quadtree	2.6%	88.1%

The robustness, efficiency and generalization power of LVC method for 3D face recognition has been proved in [4]. The main object of this paper is to learn the efficient codes for constructing the codebook. The clustering centers of different methods are shown in Fig.4. The data points adopted in Fig.4(a) are from the 2-dimensional Gabor filter response vectors of 100 training 3D faces, which contains the high-density data area, low-density data area and noisy data area. Based on the three rules in Section.2, we find that:
(1) Ncut and Mshift give too many centers to the low-density data area, which breaks Rule2. Because each learned center has the same weight when classification, this will reduce the discriminative power of the learned centers.
(2) G-means and K-means assign some centers to the data points distributed in the margin area, which will introduce some errors to the learned centers.
(3) Quadtree clustering gives most efficient centers according to the three rules, which accurately reflect the original training data distribution. Therefore, it gives the best recognition performance in our experiments.

4. CONCLUSION

In this paper, we have proposed a Quadtree clustering method to learn the efficient codes for constructing the Visual Codebook. Quadtree is a kind of hierarchical clustering, which can adaptively assign the learning centers according to the density of training data distribution and robustly overcome the influence of noisy training data. Experimental results illustrate the representing and discriminative power of the Quadtree clustering centers for 3D face recognition.

Acknowledgement

This work is supported by the National Basic Research Program of China (Grant No. 2004CB318100), the National Natural Science Foundation of China (Grant No. 60736018, 60335010, 60702024, 60723005), the National Hi-Tech Research and Development Program of China (Grant No.2006AA01Z193, 2007AA01Z162), and the Chinese Academy of Sciences.

5. REFERENCES

[1] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[2] A. Agarwal and B. Triggs, "Hyperfeatures: Multilevel local coding for visual recognition," *European Conference on Computer Vision*, 2006.

[3] Xin Meng, Shiguang Shan, Xilin Chen, and Wen Gao, "Local visual primitives (lvp) for face modelling and recognition," *International Conference on Pattern Recognition*, vol. 2, pp. 536–539, 2006.

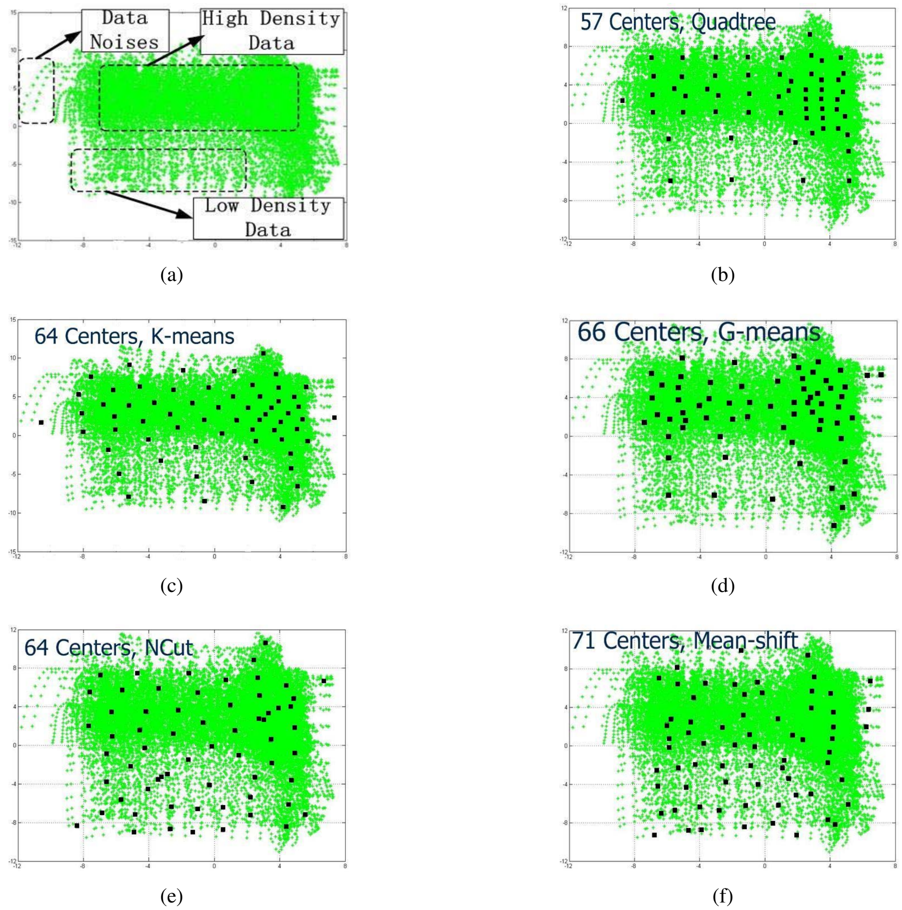


Fig. 4. A comparison of clustering centers from different methods. (a) shows the facial data distribution of the 2-dimensional Gabor filter response vectors; (b)(c)(d)(e)(f) show the clustering centers from Quadtree clustering, Kmeans clustering, Gmeans clustering, Normalized-cut clustering and Mean-shift clustering respectively.

[4] Cheng Zhong, Zhenan Sun, and Tieniu Tan, "Robust 3d face recognition using learned visual codebook," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[5] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," *IEEE International Conference on Computer Vision*, vol. 1, pp. 604–610, 2005.

[6] Tinne Tuytelaars and Cordelia Schmid, "Vector quantizing feature space with a regular lattice," *IEEE International Conference on Computer Vision*, 2007.

[7] Greg Hamerly and Charles Elkan, "Learning the k in k-means," in *Proc. NIPS*, 2003.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, Second Edition*, John Wiley & Sons, Inc., 2001.

[9] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 8, 2000.

[10] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach towards feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 5, 2002.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, , no. 3, 1999.

[12] Alan Watt, *3D Computer Graphics, 3rd Edition*, Addison-Wesley.

[13] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.