3D Face Expression Recognition Based on Histograms of Surface Differential Quantities

Huibin LI[1,2], Jean-Marie MORVAN[1,3,4], and Liming CHEN[1,2]

[1]Université de Lyon, CNRS
[2]Ecole Centrale de Lyon, LIRIS UMR5205, F-69134, Lyon, France
[3]Université Lyon 1, ICJ UMR5028, F-69622, Lyon, France
[4]King Abdullah University of Science and Technology, GMSV Research Center, Bldg 1, Thuwal 23955-6900, Saudi Arabia
{huibin.li,liming.chen}@ec-lyon.fr
morvan@math.univ-lyon1.fr

Abstract. This paper presents a novel mesh-based method for 3D face expression recognition using a local shape descriptor and the SVM classifier. To characterize shape information of the local neighborhood of facial landmarks, we calculate the weighted statistical distributions of surface differential quantities, including histogram of mesh gradient (HoG) and histogram of shape index (HoS). Normal cycle theory based curvature estimation method is employed for the first time on 3D face models. Meanwhile, the commonly used cubic fitting curvature estimation method is also employed to get the comparable results. Based on the basic fact that different expressions involve in different local shapes (shape deformation), the SVM classifier with both linear and RBF kernels achieve state of the art results on the subset of BU-3DFE database with the same experimental setting.

Keywords: 3D facial expression recognition, normal cycle theory, curvature tensor, histogram of surface differential quantities, SVM classifier

1 Introduction

Facial expression recognition (FER) is attracting a great deal of attention because of its usefulness in many applications such as human-computer interaction and the analysis of conversation structure [1]. Ekman et al. [2], who are pioneers studying human facial expressions, introduced the Facial Action Coding System (FASC) and gave evidence to classification prototypical facial expressions as happiness, sadness, anger, fear, surprise, disgust and neutral. Inspired by FASC, many researchers dedicated to FER firstly in static 2D images and dynamic 2D videos [3] and more recently on static 3D scans [4,5,6,7,8,9,10,11] and dynamic 3D videos [12,13]. With the great progress of 3D data acquisition equipments, the use of 3D facial data for FRE has attracted more attention as it contains geometry information closely sensitive to expression variations, and further on it is invariant to pose and illumination changes.

In general, existing methods for FER based on static 3D data can be categorized into two streams, i.e. feature based and model based. The first category claims that the distributions of facial surface geometric informations such as gradient and curvature [5], distances between pairs of interest landmarks [6] and local shapes near landmarks [10,11], are closely related to expression categories. These geometric informations are then extracted as features for different classifiers such as linear discriminant analysis (LDA), support vector machine (SVM) or Neural Network etc., to get the final classification results. The main drawback of this kind of approaches is that they need manual labeled landmarks. The second category tries to simulate the physical process of generating expression and explores a generic elastically deformable face model, which can generate universal expressions by adjusting parameters [7]. In general, this kind of methods needs alignment and normalization steps to find one-to-one correspondence among 3D faces, then the shape deformations between each pair of faces can be represented by model parameters [7] or feature vectors [9] etc, which are further used to perform FER. The main advantage of this kind of approaches is that they work without manual assistance.

Recently, the traditional image based 2D SIFT [14](Scale Invariant Feature Transform) framework has been extended to range image based 2.5D SIFT [15] and discrete surface based 3D mesh-SIFT [16]. Different from 2D SIFT computing gradient, both 2.5D and mesh-SIFT extracted surface normal and curvature (coded by shape index) on the local patches of detected salient points. The histograms of these differential quantities are then used as local shape descriptors for further surface matching. It has shown that mesh-SIFT is efficient to 3D face recognition [16]. Inspired by the success of 3D mesh-SIFT, in this paper, we try to exploit mesh-SIFT framework for facial expression recognition. Different from mesh-SIFT, the manual landmarks round the regions of eyebrows, eyes, nose and mouth are chosen as salient points. Within each local neighborhood of landmark, we calculate the weighted statistical distributions of surface differential quantities, including histogram of mesh gradient (hog^p) and histogram of shape index (hos^p). The features for each 3D face model can be obtained by concatenating hog^p or hos^p of all the selected landmarks with a fixed order. These features are then classified by SVM to achieve facial expression classification. Curvatures of the discrete face models are estimated using normal cycle theory based method and local cubic fitting method on mesh data. The experimental results on the subset of BU-3DFE database highlight the effectiveness of the proposed method.

The remainder of this paper is organized as follows: differential quantities estimated on mesh-based facial models are introduced in section 2, and section 3 presents the local shape descriptors. Experimental results are discussed in section 4. Section 5 concludes the paper.

2 Estimating Differential Quantities on Triangular Meshes

Since we will base our descriptors on shape index and mesh gradient, we need to estimate the curvatures and normal vectors firstly. In this section, we describe two curvature estimation methods: normal cycle theory based method proposed by Cohen-Steiner and Morvan; local cubic fitting methods proposed by J. Goldfeather and V. Interrante. Then, we give the method to compute mesh gradient.

2.1 Estimating Curvature by Normal Cycle Theory based Method

There are many approaches to calculate curvature on triangular meshes based on estimation of curvature tensor. Taubin [17] introduced a 3D curvature tensor from which the principal curvatures directions can be estimated by two of the three eignvectors and the principal curvature can be computed by linear combinations of two of the three eigenvalues. Cohen-Steiner and Morvan [18,19] also given discrete definition for the mean, Gaussian curvature and the curvature tensor based on normal cycle theory, and prove that the estimated curvature tensors converge to the true ones of the smooth surface under specific sampling conditions. The basic idea and its discrete from can be carried out as follows [20]:

For every edge e of the mesh, there is an obvious minimum (i.e., along the edge) and maximum (i.e., across the edge) curvature. A nature curvature tensor can therefore be defined at each point along an edge, named as generalized curvatures [18]. This line density of tensors can now be integrated over the arbitrary region B by summing the different contributions from B, leading to the simple expression:

$$\mathcal{T}(v) = \frac{1}{B} \sum_{edges\, e} \beta(e) |e \cap B| \, \overline{e}\, \overline{e}^t \tag{1}$$

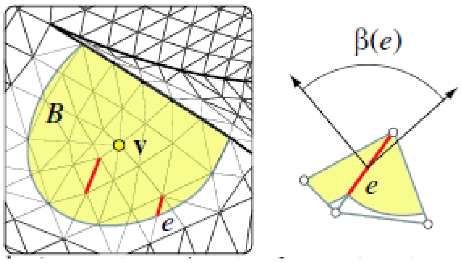

Fig. 1. Illustrated of normal cycle theory based curvature estimation method (equation (1) [20]).

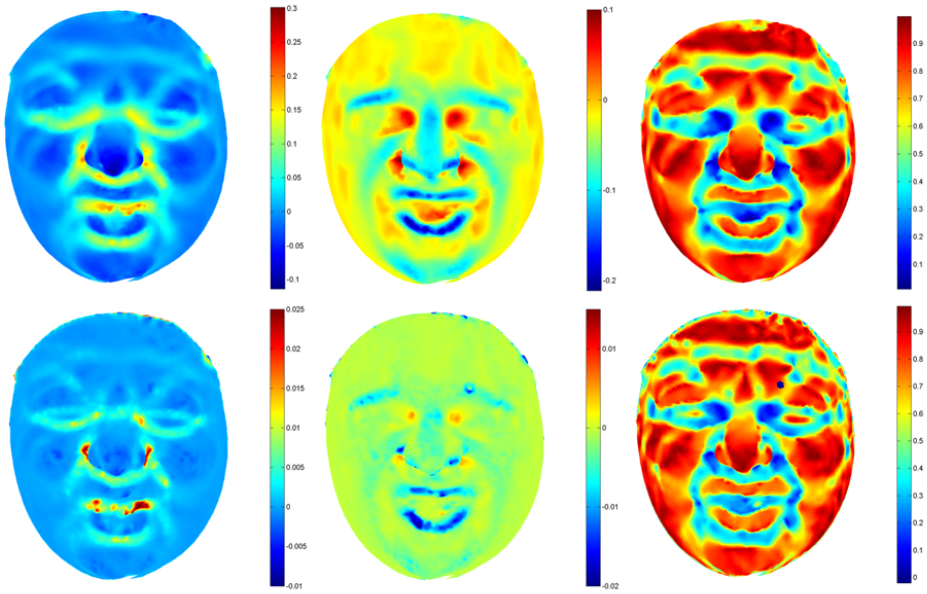

Fig. 2. First row, from left to right: κ_{max}, κ_{min} and shape index estimated by normal cycle theory based method. Second row, from left to right: κ_{max}, κ_{min} and shape index estimated by cubic fitting based method, (models of M0044-DI03).

Where v is an arbitrary vertex on the mesh, $|B|$ is the surface area around v over which the tensor is estimated. $\beta(e)$ is the signed angle between the normals to the two oriented triangles incident to edge e (positive if convex, negative if cancave), $|e \cap B|$ is the length of $e \cap B$ (always between 0 and e), and $\overline{e}$ is a unit vector in the same direction as e. In our experiments, we estimate the tensor at every vertex location v, for a neighborhood B of 2-ring. The principal curvatures k_{min} and k_{max} at v can now be estimated by the two maximum eigenvalues of $\mathcal{T}(v)$. Fig. 1 shows the schematic of this method.

Shape index which expressing different shape classes by one single number from 0 to 1 then can be estimated by the following equation:

$$\mathbf{S} = \frac{1}{2} - \frac{1}{\pi} arctan(\frac{\kappa_{max} + \kappa_{min}}{\kappa_{max} - \kappa_{min}}) \tag{2}$$

The first row of Fig.2 shows a example of maximum, minimum curvatures and shape index estimated by this method on a 3D face model.

2.2 Estimating Curvature by Local Cubic Fitting based Method

In order to compare the performances of our shape descriptors based on different curvature estimation methods used for facial expression classification, we also adopt local cubic fitting method [20] to estimate curvatures. The basic idea of

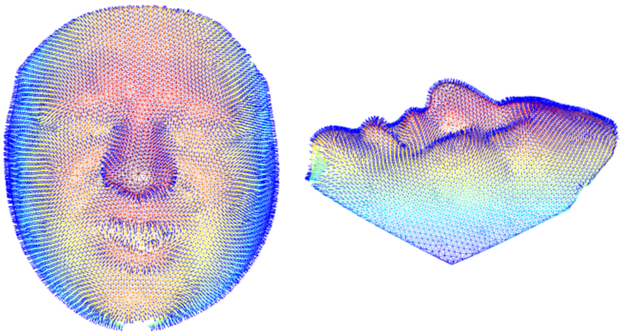

Fig. 3. Illustrated the estimated normal vectors (models of M0044-DI03).

this method is that: for each vertex p of the 3D mesh, a local coordinate system is defined by taking the vertex p as an origin and normal vector $n_p = (n_x, n_y, n_z)^T$ as the z axes. Two orthogonal axes, x and y, randomly chosen in the tangent plane perpendicular to the normal vector. The local neighborhood points (2-ring in our paper) and its corresponding normal vectors are first transformed to the local coordinate system, then used for fitting a cubic function and its normal respectively. The cubic function and its normal having the following forms:

$$z(x,y) = \frac{A}{2}x^2 + Bxy + \frac{C}{2}y^2 + Dx^3 + Ex^2y + Fxy^2 + Gy^3 \tag{3}$$

$$\begin{aligned}(z_x, z_y, -1) = (&Ax + By + 3Dx^2 + 2Exy + Fy^2 + \\ &+ Bx + Cy + Ex^2 + 2Fxy + 3Gy^2, -1)\end{aligned} \tag{4}$$

By using least-square fitting method to solve the fitting equations (3) and (4), the Weingarten matrix on a vertex can be computed as:

$$\mathbf{W} = \begin{pmatrix} \frac{\partial^2 z(x,y)}{\partial x^2} & \frac{\partial^2 z(x,y)}{\partial x \partial y} \\ \frac{\partial^2 z(x,y)}{\partial x \partial y} & \frac{\partial^2 z(x,y)}{\partial y^2} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{5}$$

The maximum curvature k_{max} and minimum curvature k_{min} then can be estimated as the eigenvaules of the Weingarten matrix. The second row of Fig.2 shows a example of maximum, minimum curvatures and shape index estimated by cubic fitting method on the same 3D face model.

2.3 Mesh Gradient Estimation

Let the normal vector at p as $n_p = (n_x, n_y, n_z)^T$, which can be estimated by average the normal vectors of one-ring faces. According to (3) and (4), the gradient direction and the gradient magnitude can be estimated as follows:

$$\theta = arctan(\frac{n_y}{n_x}) \tag{6}$$

$$||\nabla z(x,y)|| = \sqrt{(-\frac{n_x}{n_z})^2 + (-\frac{n_y}{n_z})^2} \tag{7}$$

Fig. 3 shows the estimated normal vectors on a 3D mesh-based face model.

3 Local Shape Descriptors

3.1 Landmarks Selection

Since we conducted our experiments on the BU3D-FE database, we extracted our local shape descriptors on the first 60 manual landmarks round of eyebrows, eyes, nose and mouth, which are selected from 83 manual landmarks in the database. For each landmarks, a neighborhood with a geodesic disk is considered. The radius of the disk is equal to 22 mm in our experiments. 60 selected landmarks and one example of local neighborhood points of left mouth corner are shown in Fig. 4.

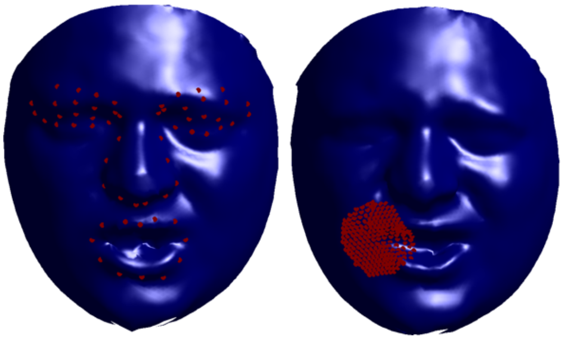

Fig. 4. From left to right, 60 selected manual landmarks, local neighborhood points of the left mouth corner (M0044-DI04 and M0044-DI03).

3.2 Local Coordinate System Obtained and Orientation Assignment

In practice, we first transform the local neighborhood points to the local coordinate system, in which the landmark point is the origin and its normal vector is along the positive z axis. Two perpendicular vectors x and y axis are randomly chosen in the tangent plane. In order to make the descriptor invariant to rotation, each landmark point is assigned one or several canonical orientations according to the dominant direction(s) of gradients in the local tangent plane with 360 bins. Once the canonical orientations are assigned, the local coordinate system rotates in the local tangent plane, making each canonical orientation as new x axis. Now y axis can be computed by cross product of z and x.

In this new local coordinate system, we project all the neighbors of a landmark point to its tangent plane. Eight projected points along to eight quantized directions starting from canonical orientation with a distance of r_1 to the landmark point are fixed. Nine circles centered at the landmark point and its eight neighbors with a radius r_2 can be further located. Fig.5 shows this arrangement.

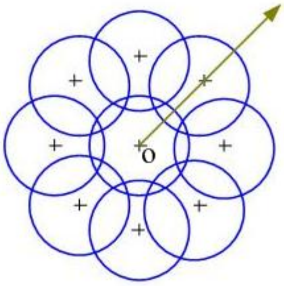

Fig. 5. Canonical orientation (arrow), landmark point (o) and its 8 neighborhood vertices (+) assigned with 9 circles.

3.3 Feature Vector Computed by Histograms of Surface Differential Quantities

In each circle, we calculate histogram of surface gradient (hog^c) and histogram of shape index (hos^c). For hog^c, we compute histogram of gradient angle weighted by gradient magnitude. This histogram is with 8 bins representing 8 main orientations ranging form 0 to 360 degree. For hos^c, the values of shape index ranging from 0 to 1 are also quantized to 8 bins. Then, all the values of histograms are weighted by Gaussian with the Euclidian distance to the center point of the circle as the standard deviation. Every histogram with the length of 72 is then normalized and form a feature vector for a single landmark point. Noted as hog^p and hos^p which are formed as follows:

$$hog^p = (hog_1^c, hog_2^c, \dots, hog_9^c) \tag{8}$$

$$hos^p = (hos_1^c, hos_2^c, \dots, hos_9^c) \tag{9}$$

The final feature vectors of each face model can be obtained by simply concatenating the histogram of all 60 manual landmarks with a fixed order respectively (expressed by HoG and HoS). They have the same length of 60 × 72 = 4320 and can be represented as follows:

$$HoG = (hog_1^p, hog_2^p, \cdots, hog_{60}^p) \tag{10}$$

$$HoS = (hos_1^p, hos_2^p, \cdots, hos_{60}^p) \tag{11}$$

The fusion of them HoG+HoS can be obtained by simply concatenating them.

4 Experimental Results

We implemented experiments on the BU-3DFE database [4] which contains 2500 textured 3D face models of 100 subjects with different gender, race, and age. Each subject contains one neutral model and six universal non-neutral expressions: happiness, sadness, anger, fear, surprise and disgust. Also, each non-neutral expression is obtained at four different gradations: low, middle, high and highest. Fig.5 shows some samples of the database with six different expressions of high and highest gradations.

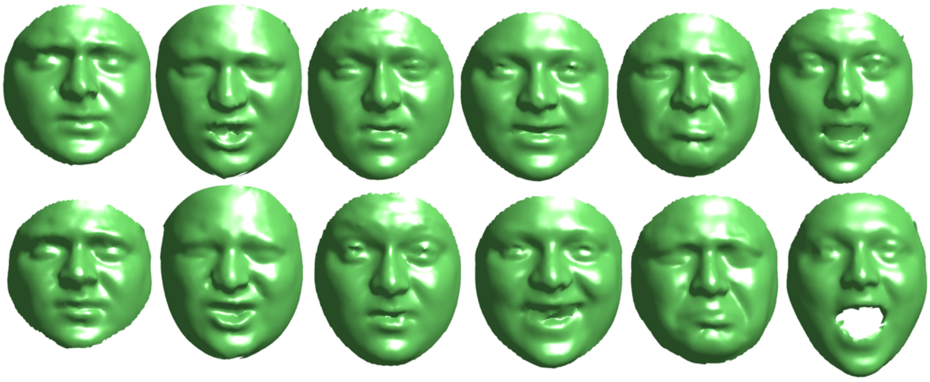

Fig. 6. Examples of six universal expressions, from left to right: anger, disgust, fear, happiness, sadness, surprise. First row, high gradation, Second row, highest gradation.

In our experiments, we use the same set up as in [9]. A subset of 60 subjects was randomly selected with two high-intensity models for each of the six facial expressions. Totally 60 × 12 = 720 3D mesh-face models were selected. Then, 54 and 6 subjects were randomly selected as training set (648 models) and test set (72 models) respectively. In practice, since the number of models in test set is very limited (only 12 models for each expression), at the same time, people of different race, gender and age may have different facial surface changes when they do the same expression, the average recognition accuracy obtained by 10 or 20 random experiments varies greatly, from about 50% to more than 90% [9] with the same feature, classifier and parameters set up. To obtain stable average recognition recognition accuracies, we run all of our experiments 1000 times independently. To calculate descriptors, we set r_1 and r_2 equal to 15 mm and 7 mm respectively. The SVM classifier with liner kernel and RBF (radial basis function) kernel is used, here the implement of LIBSVM [22] is employed. The parameter 'gamma' for RBF kernel is chosen as 0.04 by 8-fold cross-validation. The results are shown in table 1-5. AN, DI, FE, HA, SA, SU are short of anger, disgust, fear, happiness, sadness, and surprise, respectively.

Table 1 shows the average confusion matrix obtained by HoG feature and SVM classifier with linear and RBF kernels. we can find that both liner and RBF kernels get high classification rates for expressions of happiness and surprise but low for other expressions. As the phenomenon obtained by other work, anger and fear have comparatively lower classification rates. Anger is confused mainly by sadness and fear is confused mainly by happiness and surprise. Linear kernel works better for anger while a little worse for disgust and happiness than RBF kernel. Both of these two kernels get almost the same average recognition rate about 76.5% for all six expressions.

Table 1. Average confusion matrix obtained by HoG

	linear kernel						RBF kernel					
%	AN	DI	FE	HA	SA	SU	AN	DI	FE	HA	SA	SU
AN	**74.0**	8.3	0.8	1.6	15.2	0.1	**66.0**	12.0	3.0	0.9	18.1	0
DI	5.9	**76.5**	10.4	3.7	2.6	0.9	2.3	**80.7**	8.5	3.8	3.2	1.5
FE	5.4	11.7	**63.6**	9.7	5.4	4.2	3.4	8.4	**63.2**	13.2	5.5	6.2
HA	1.1	0.9	16.0	**82.0**	0	0	0.1	1.4	12.6	**85.6**	0	0.3
SA	16.6	2.7	8.2	0	**72.2**	0.3	15.6	4.1	5.7	0	**72.6**	2.0
SU	1.0	2.3	4.9	0.3	1.5	**90.0**	0	2.6	4.1	1.6	0.9	**90.7**
Average	**76.4**						**76.5**					

Table 2. Average confusion matrix obtained by normal cycle based HoS

	linear kernel						RBF kernel					
%	AN	DI	FE	HA	SA	SU	AN	DI	FE	HA	SA	SU
AN	**77.0**	8.5	0.9	0	13.3	0.4	**71.5**	10.4	2.2	0	15.2	0.7
DI	7.6	**80.0**	6.2	2.8	3.4	0	3.6	**82.0**	6.5	3.4	4.4	0
FE	5.2	5.9	**70.9**	8.9	5.9	3.2	4.4	8.1	**65.6**	12.0	6.1	4.0
HA	0.4	0.9	4.8	**93.2**	0	0	0.6	1.0	6.5	**91.2**	0	0.7
SA	15.5	1.8	7.3	0	**74.4**	0.9	14.1	2.4	10.1	0	**72.0**	1.5
SU	0.2	1.9	1.9	0.4	0	**95.6**	0.1	1.6	1.4	0.2	0	**96.8**
Average	**81.9**						**79.9**					

Table 2 shows the average confusion matrix obtained by HoS feature (estimated by normal cycle theory based method) and SVM classifier of linear and RBF kernels. All the results are better than the corresponding results in Table 1 except the one obtained by RBF kernel for sadness (72%, vs 72.6%). Linear kernel works much better than RBF kernel except disgust (80.0% vs 82.0%) and surprise (95.6% vs 96.8%). The average recognition rates for all six expressions are 81.9% and 79.9% for linear and RBF kernals.

Table 3 shows the average confusion matrix obtained by HoS feature (estimated by local cubic fitting method) and SVM classifier of linear and RBF kernels. All the results are better than the corresponding results in table 1 except the one obtained by linear kernel for anger (73.1%, vs 74.0%). For the case of linear kernel, the performances are worse than the ones in table 2 except for disgust (80.6% vs 80.0) and fear (72.3% vs 70.9%); while for the case of RBF kernel, the performances are a litte better than table 2 (80.6% vs 79.9) especially for sadness (76.2% vs 72%). Two curvature estimation methods get comparative results.

Table 3. Average confusion matrix obtained by cubic fitting based HoS

	linear kernel						RBF kernel					
%	AN	DI	FE	HA	SA	SU	AN	DI	FE	HA	SA	SU
AN	**73.1**	7.7	2.7	0.2	16.2	0	**72.3**	9.9	2.6	0	15.2	0
DI	5.1	**80.6**	9.3	3.0	1.7	0.5	3.3	**81.3**	8.5	2.9	2.7	1.2
FE	5.4	4.8	**72.3**	8.7	6.0	2.7	5.5	73.3	**66.6**	9.5	6.7	4.4
HA	0.8	3.9	4.2	**90.5**	0	0.7	0.9	2.2	5.2	**91.1**	0	0.7
SA	16.3	0.6	5.9	0	**76.3**	0.9	12.8	1.5	8.6	0	**76.2**	0.9
SU	0.1	2.3	3.0	0	0.8	**93.9**	0.1	2.3	1.1	0	0.6	**95.9**
Average	**81.1**						**80.6**					

Table 4. Average confusion matrix obtained by HoG+HoS descriptor using linear kernel of SVM

	normal cycle based						cubic fitting based					
%	AN	DI	FE	HA	SA	SU	AN	DI	FE	HA	SA	SU
AN	**76.8**	7.6	2.1	0	13.5	0	**76.4**	8.0	1.8	0	13.7	0
DI	7.6	**78.1**	6.6	2.1	5.0	0.7	4.4	**80.2**	10.2	2.8	2.0	0.5
FE	4.6	7.6	**73.2**	7.3	5.1	2.3	5.1	6.2	**73.6**	8.0	5.3	1.7
HA	0.5	0.5	6.8	**91.4**	0	0.8	0.8	2.2	6.5	**90.4**	0	0.8
SA	14.5	1.1	8.3	0	**75.5**	0.6	14.7	0.5	6.2	0	**77.8**	0.8
SU	0	1.7	2.0	0.9	0.8	**94.5**	0	2.0	3.3	0.1	1.0	**93.6**
Average	**81.6**						**82.0**					

Table 4 presents the average confusion matrix obtained by the fusion feature HoG+HoS (estimated by normal cycle theory and cubic fitting) and SVM classifier of linear kernel. Compared to the results in table 2 left and table 3 left (linear kernel), the results drop down except fear (73.2% vs 70.9%) and sadness (75.5% vs 74.4%) in the case of normal cycle theory method, while the results increase for anger, fear, and sadness (especially for anger (76.4% vs 73.1%)) and drop down only a little for other expressions in the case of cubic fitting method. But there are no increase (normal cycle) or a little increase (cubic fitting) of the average recognition rates for all six expressions. It may be caused by the huge dimension (8640×1) of the fused descriptor.

Table 5 gives the results of our method (HoG by RBF kernel, HoS by normal cycle and linear kernel, HoG+HoS by cubic fitting and linear kernel) and the ones reported in [9] and [11]. In fact, in Gong et al. (Gong) [9], the results of the approaches in Wang et al. (Wang) [5], Soyel et al. (Soyel) [6], and Tang et al.

(Tang) [8], are obtained on the same experimental setting. While in Berretti et al. (Berretti) [11], 60 subjects are selected randomly from experiment to experiment. It can be found that our approach using HoG feature obtained a comparative result to others, further on, the results obtained by HoS and HoG+HoS features outperform all the other methods.

Table 5. Average expression recognition rats for our method and the works in [11], [9], [5], [6], [8].

	HoG	HoS	HoG+HoS	Berretti	Gong	Wang	Soyel	Tang
AVE	**76.48%**	**81.86%**	**82.01%**	**77.54%**	**76.22%**	**61.79%**	**67.52**	**74.51%**

5 Conclusions and Future Work

In this paper, we have developed a mesh-based 3D facial expression recognition approach and evaluated it on the subset of BU-3DFE database. Our novel approach is based on a local shape descriptor (HoG and HoS) computed from surface differential quantities and the SVM classifier. The surface differential quantities are extracted on the local neighborhoods of 60 manual landmarks. Curvatures are estimated by normal cycle theory based method and cubic fitting method. Both linear and RBF kernels of SVM are employed for classification. The results indicated that both curvature estimation method and both linear and RBF kernels get nearly the same average expression recognition rates. There is no single descriptor or kernel which works best for all the six expressions. Similar to what Gong et al. said in [9], anger, fear and sadness are more difficult than disgust, happiness and surprise for all the descriptors and kernels.

For the future work, we will test our method on the whole BU3D-FE database and consider the other two lower expression gradations. Further on, less training set and more test set (50% vs 50%) will also be considered . At last, well analyzing and understanding the deep relations between different expression (principal of their transfer) can help us to improve the results of expression recognition.

References

1. Otsuka, K., Sawada, H., Yamato, J., C.: Automatic inference of cross-modal nonverbal interactions in multiparty conversations: "who responds to whom, when, and how?" from gaze, head gestures, and utterances. In: ICMI, pp. 255–262, ACM Press, New York, (2007).
2. Ekman, P., C.: Universals and cultural differences in facial expressions of emotion. In: In Nebraska Symposium on Motivation, pp. 207 –283, Lincoln, NE, (1972).
3. Fasel, B., Luettin.J., C.: Automatic facial expression analysis: a survey. Pattern Recognition, 259–275,(2003)
4. L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. C.: A 3D Facial Expression Database For Facial Behavior Research. In: The 7th International Conference on

Automatic Face and Gesture Recognition (FG). pp. 211–216, IEEE Computer Society TC PAMI, (2006)
5. J. Wang, L.Yin, X. Wei, and Y. Sun, C.: 3d facial expression recognition based on primitive surface feature distribution, In: CVPR, pp. 1399–1406, (2006).
6. H. Soyel and H. Demirel. C.: Facial expression recognition using 3d facial feature distances. In: In Int. Conf. on Image Analysis and Recognition, pp. 831–838, (2007).
7. I. Mpiperis, S. Malassiotis, and M. G. Strintzis. S.: Bilinear models for 3-d face and facial expression recognition. IEEE Transactions on Information Forensics and Security,3(3): 498–511,(2008).
8. H. Tang and T. S. Huang. C.: 3d facial expression recognition based on automatically selected features. In: In Int. Conf. on Computer Vision and Pattern Recognition, pp. 1–8, (2008).
9. B. Gong, Y. Wang, J. Liu, and X. Tang. C.: Automatic facial expression recognition on a single 3d face by exploring shape deformation. In: In Int. Conf. on Multimedia, pp. 569–572, (2009).
10. A. Maalej, B. Ben Amor, M. Daoudi, A. Srivastava, S. Berretti. C.: Local 3D Shape Analysis for Facial Expression Recognition. In: ICPR, pp. 4129–4132, IEEE, (2010).
11. S. Berretti, A. D. Bimbo, P. Pala, B. Ben Amor, M. Daoudi. C.: A Set of Selected SIFT Features for 3D Facial Expression Recognition. In: ICPR, pp. 4125–4128, IEEE, (2010).
12. L. Yin, X. Chen, Y. Sun, Worm, T., Reale, M., C.: A High-Resolution 3D Dynamic Facial Expression Database. In: IEEE Int. Conference on Automatic Face Gesture Recognition, pp. 1–6, (2008).
13. Sun, Y. and Yin, L. C.: Facial Expression Recognition Based on 3D Dynamic Range Model Sequences. In: 10th European Conference on Computer Vision. pp. 58–71, Springer-Verlag, Berlin, 2008.
14. D.G.Lowe, S.: Distinctive image features from scale invariant keypoints. IJCV. pp.91–110, (2004).
15. T.W. Rachel Lo and J. Paul Sieberta, C.: Local feature extraction and matching on range images: 2.5D SIFT, In: Computer Vision and Image Understanding. Vol 113(12), pp. 1235-1250, (2009)
16. M.C.Fabry et al, C.: Feature detection on 3d face surfaces for pose normalisation and recognition, In: BTAS, (2010).
17. Taubin, G. C.: Estimating the tensor of curvature of a surface from a polyhedral approximation, In: ICCV, pp. 902–, IEEE Computer Society, USA, (1995).
18. Cohen-Steiner, David. and Morvan, J.M. C.: Restricted delaunay triangulations and normal cycle In: Proceedings of the nineteenth annual symposium on Computational geometry. pp. 312–321, ACM, New York, (2003)
19. J.M. Morvan. C.: Generalized Curvatures. Springer-Verlag, Berlin (2008)
20. P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, M. Desbrun. S.: Anisotropic polygonal remeshing. ACM Trans. Graph. 22(3),485–493, (2003).
21. J. Goldfeather and V. Interrante, S.: A novel cubic-order algorithm for approximating principal direction vectors. ACM Trans. Graph. pp.45–63, (2004).
22. Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm.`