Landmark Localization on 3D/4D Range Data Using a Shape Index-Based Statistical Shape Model with Global and Local Constraints

Shaun Canavan*, Peng Liu, Xing Zhang, and Lijun Yin

Department of Computer Science, State University of New York at Binghamton, Binghamton, NY, 13902, USA

Abstract

In this paper we propose a novel method for detecting and tracking facial landmark features on 3D static and 3D dynamic (a.k.a. 4D) range data. Our proposed method involves fitting a shape index-based statistical shape model (SI-SSM) with both global and local constraints to the input range data. Our proposed model makes use of the global shape of the facial data as well as local patches, consisting of shape index values, around landmark features. The shape index is used due to its invariance to both lighting and pose changes. The fitting is performed by finding the correlation between the shape model and the input range data. The performance of our proposed method is evaluated in terms of various geometric data qualities, including data with noise, incompletion, occlusion, rotation, and various facial motions. The accuracy of detected features is compared to the ground truth data as well as to start of the art results. We test our method on five publicly available 3D/4D databases: BU-3DFE, BU-4DFE, BP4D-Spontaneous, FRGC 2.0, and Eurecom Kinect Face Dataset. The efficacy of the detected landmarks is validated through applications for geometric based facial expression classification for both posed and spontaneous expressions, and head pose estimation. The merit of our method is manifested as compared to the state of the art feature tracking methods.

1. Introduction

Applications such as face recognition, expression analysis, human-computer interaction, and face video segmentation are increasingly being developed based on 3D, and 4D (3D+time) range data [34][36][38] [41][42][43][44], given the rapid technological advancement of 3D imaging systems [37][40][16] [46]. Landmark localization on 3D/4D range data is the first step toward geometric based vision research for object modeling, recognition, visualization, and scene understanding [35][39][47][48][49]. Landmark localization is a crucial task when dealing with 3D and 4D data. For example it allows for simultaneous location of multiple objects in a scene.

While 2D based tracking methods have been successfully developed, such as Active Shape Models [7], Active Appearance Models [18], using a consensus of exemplars [2], Constrained Local Models (CLM) [9], regularized landmark mean-shift [24], generative shape regularization model [13], explicit shape regression [6], supervised descent method for face alignment [28], and shape-constrained linear predictors [45], there is a need for novel and robust algorithms to handle 3D/4D range data. Morphable Model [4][53] is one of the successful algorithms for handling 3D range data.

* Corresponding Author: shaun.canavan@binghamton.edu (Shaun Canavan)*, lijun@cs.binghamton.edu (Lijun Yin)

There has been recent work to address the problem of detecting feature landmarks on range data. Zhao et al. [32] had success with detecting 3D landmarks using a statistical facial feature model; however there is an upper bound on the number of landmarks. Fanelli et al. [11] used an active appearance model that is based on random forests; however this method used depth and intensity data rather than the 3D/4D range data. Sun et al. [25] used a so-called vertex flow approach, which used an active appearance model (AAM) to track features of 3D range models. However, the tracking of facial features was not truly in the 3D space, rather it was tracked in the 2D space and the 3D features themselves were obtained by mapping the 2D features to the corresponding parts of the 3D models, tending to cause inaccurate projections. Nair et al. [20] fit a 3D active shape model to facial data using candidate landmarks to deform the model, however the resulting error rate for fitting is relatively large, and problems occur when holes exist around the nose. Zhou et al. [33] created a 3D active shape model which was trained using a 3DMM, although the fitting for this method was done in 2D. Perakis et al. [21] used a 3D active shape model which was fit from previously determined candidate landmarks. A draw-back to this method is the need for preprocessing. Guan et al. [14] performed landmark localization on facial data by utilizing a Bezier surface. This method was tested on a small dataset consisting of 100 3D models. Jeni et al. [15] used a 3D constrained local model method (estimated from 2D shape) to track landmarks for action unit intensity estimation. Baltrisaitis [1] used a 3D CLM (a.k.a. CLM-Z) trained with depth data rather than 3D/4D range data for rigid and non-rigid feature tracking. A statistical model (blend-shape) was utilized by Weise et al. [27] to track facial data and animate a virtual avatar; however, this has the limitation that the blend-shape may not have a unique set of needed weights for an expression. Chen et al. [39][47] applied a coarse-to-fine approach via curvature and active normal model for landmarking. Bonde et al. compute the shape-index of discrete points, on 2.5D data, to recognize objects [50]. Chen et al. compute the shape-index along with histograms to recognize 3D data [51]. Wang et al. use shape-index and geodesic distances to find correspondences between 3D objects [52]. Recently, we have developed a so-called 3D temporal deformable shape model (TDSM) for feature tracking through 3D range sequences [5]. However, such a multi-frame based shape model may not work well for different expressions within a very short duration when dramatic motions or sudden expression changes occur in the 3D videos, thus the performance on 3D geometric tracking still needs to be improved. Motivated by the previous work [5], we continue to address the issue of feature detection and tracking on 3D/4D range data with a more reliable way.

In this paper, we study the challenges of detecting and tracking landmarks by proposing to construct a *shape index-based statistical shape model (SI-SSM) with both global and local constraints*. The SI-SSM is constructed from both the global shape of 3D feature landmarks and local features from patches around each landmark. In order to construct the patches we find 3D features from the (u, v) coordinates around each landmark. From these new features we construct a $n \times n$ patch, where each vertex is represented by a unique shape index value. Using both the global shape and the local features around each landmark enables us to reliably detect and track features on the range mesh data. The feature detection and tracking are based on finding the correlation between the local shape index patches on the input range data and the trained SI-SSM model (as illustrated in Figure 3).

The main contribution of this paper is the construction of a statistical model that makes use of both the global shape of 3D face surface, as well as the local shape around individual features by way of shape index representation. This model can be used to detect and track features on range data. By using the shape index representation we are able to make the local fitting invariant to both lighting and pose changes. We are able to model and fit data that includes various emotions, rotations, occlusions, and missing data by training on each of these data types. Following is the summary of the main contribution of this work:

(1) We proposed and developed a novel approach for 3D/4D facial feature detection and tracking. This approach has extended the global statistical shape model to an integrated global and local shape model to improve the tracking performance with respect to various imaging data conditions. In particular, we have presented a shape-index based local shape model and combined this model with

the global shape model as a new statistical shape descriptor (so-called *shape-index based statistical shape model (SI-SSM)*).

(2) We have tested the new SI-SSM model on five public 3D/4D face databases (i.e., BU-3DFE [29], BU-4DFE [30], BP4D-Spontaenous [31], FRGC 2.0 [22], and Eurecom Kinect Face Database [19]) which cover a variety of data types, including static vs. dynamic, posed vs. spontaneous, high-resolution vs low-resolution, etc.

(3) We show the merit of the new SI-SSM based detection and tracking through performance evaluations with respect to various authentic facial behaviors, dramatic head rotations, data conditions with noise, occlusion, and incompleteness, as well as comparison with four state of the art approaches.

(4) We have validated the usability of our new approach through its application to facial expression recognition and head pose estimation. Especially, we applied a *spatial-temporal HHM model* to classify six posed expressions on 4DFE and eight spontaneous expressions on BP4D-Spontaneous database successfully.

The paper is organized as follows: Section 2 presents the new statistical model and its construction. Section 3 describes the feature detection and tracking algorithm in detail. The experiments and evaluations are reported in Section 4, followed by application study for 3D/4D face analysis. Finally, the conclusion and future work are discussed in Section 6.

2. Shape Index-based Statistical Shape Model (SI-SSM)

Our proposed method models both the global shape of 3D facial landmarks, as well as the local curvatures from patches around the landmarks. In order to construct the SI-SSM, we annotate the training data with *L* landmarks. From these annotated landmarks we are able to model both the global and local shapes of a face. An example of an annotated mesh can be seen in Figure 1, where *L=83*. The resulting global shape, local curvature patches, and the final construction of the SI-SSM are detailed in the following sub-sections.

2.1 Global Face Shape

To model the global face shape, we first create a $n \times n$ patch around each of the *L* annotated landmarks for each training mesh. To construct these patches we use the corresponding (u, v) coordinates for each of the training data. An example of a 3D mesh with patches can be seen in Figure 1.

Given a set of *M* training data, each with *L* patches, a parameterized model, S_G, is constructed. This parameterized model contains the global shape of all of the training data, where $S_G = (x_1, y_{1,} z_1, \dots, x_N, y_N, z_N)$, where $N = L \times n \times n$. The first step to create this model is aligning the *N* landmarks, on each of the *M* training data, by using a modified version of Procrustes analysis [7]. PCA is then applied to learn the modes of variation from the training data. For our experiments we keep approximately 95% of the variance. We can then approximate any shape by

$$S_G = \bar{s} + Vw \tag{1}$$

where $\bar{s}$ is the mean shape, *V* is the eigenvectors of the covariance matrix *C*, which describes the modes of variation learned from the training data, and *w* is a weight vector used to generate new shapes (referred to as an instance of the SI-SSM) by varying its parameters within certain limits. We impose these limits to ensure only valid shapes are constructed. For our model we constrain those valid shapes to be within two standard deviations from the mean (which is the allowable shape domain)

$$-2\sqrt{\lambda_i} \leq w_i \leq 2\sqrt{\lambda_i} \tag{2}$$

Where λ_i is the i[th] eigenvalue of *C*. We have empirically found, from the training data, ± 2 standard deviations from the mean to be a suitable constraint for our model as this range gives us a good balance between speed and accuracy of model fit. A smaller constraint would shrink the search space and possibly miss the best fit to the input model. A larger domain would create an unnecessarily large search space that would have instances of the model that do not look like a face.

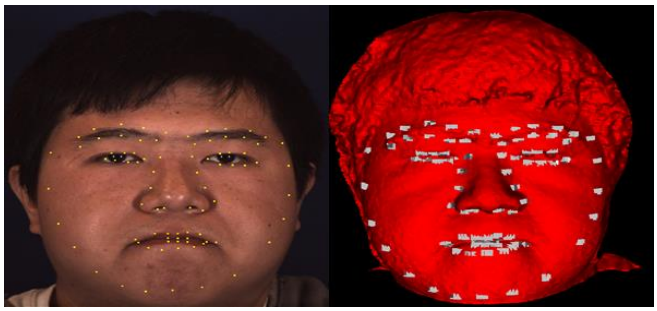

Fig. 1: Left: 83 landmarks defined on a face; Right: corresponding 3D patches with grey-scale shape index values.

2.2 Local Face Shape

To model the local face shape we apply the shape index values to represent the local patches. To do so, we calculate the shape index values for each of the *L* patches in the global face shape. Calculating the shape index gives us a quantitative measure of the shape of each patch around the *L* annotated landmarks. Shape index is defined as follows:

$$SI = \frac{2}{\pi} * arctan\left(\frac{k_2 + k_1}{k_2 - k_1}\right) \tag{3}$$

where k_1 and k_2 are the min and max principal curvatures of the surface, with $k_2 \geq k_1$. All shapes can be mapped to the range [-1.0, 1.0], where each unique shape corresponds to a specific shape index value. A cubic polynomial fitting approach is used to compute the eigen-values of the Weingarten Matrix [10] giving us k_1 and k_2. We normalize the shape index scale to [0, 1] and encode them as a continuous range of grey-level values between 1 and 255. To give us an efficient description of the data, we transform the shape index scale to a set of nine quantization values from concave to convex. Figure 2 shows example range meshes with the shape index values normalized to [1, 255] for illustration.

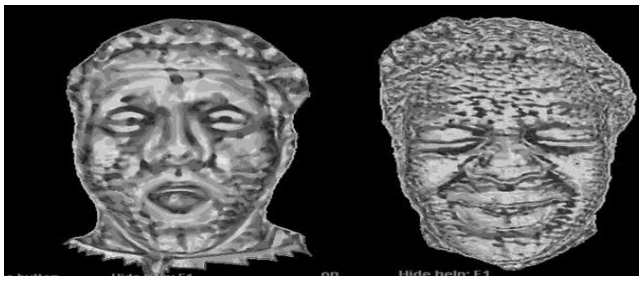

Fig. 2 Example of shape index (grey-scale) on mesh models.

Given the set of *M* training data with *L* patches where each contains the calculated shape index values, we construct a second parameterized model $S_L = (SI_1, \ldots, SI_N)$. PCA is then applied to this local shape vector in the same manner as the global shape vector does. We construct a new vector, V_{SI}, which yields of the modes of variation along the principal axes for the local shape index values. Similar to the global shape, we can approximate any local patch shape using the vector, V_{SI}, and a weight vector w_{SI} by

$$S_L = \bar{si} + V_{SI} w_{SI}.$$

2.3 Combined Global and Local Feature Model

To take both global and local shape constraints, we integrate the two features into a combined feature model. To do so, we concatenate both the global and local shape feature vectors into one feature vector $\boldsymbol{S_{GL}}$, where $\boldsymbol{S_{GL}} = (\boldsymbol{x_1}, \boldsymbol{y_1}, \boldsymbol{z_1}, \dots, \boldsymbol{x_N}, \boldsymbol{y_N}, \boldsymbol{z_N}, \boldsymbol{SI_1}, \dots, \boldsymbol{SI_N})$. Using the combined feature vector allows us to move the local patches, on the face data, to a more representative surface on the model while maintaining the constraint that we approximate a valid face shape in the allowable shape domain. Other methods that use statistical models such as [8][9][25] have been successful in using statistical models to create a combined feature vector that incorporates both the shape and "appearance" of the face. The "appearance" portion (e.g. textures) of the model helps to guide the model and fit to new data, however, these approaches suffer from the problem of global lighting variation, as well as skin tone of the modeled face. The grey-level appearance information in these models must be normalized in order to handle this lighting variation. Our SI-SSM uses shape index values to model our local features, which guide our model and fit to new range data. Shape index values are invariant to global lighting variation and skin tone. As described in the previous section, shape index is a quantitative measure of shape, so using these features our model does not encounter the same issues that similar "appearance" based solutions do.

3. 3D/4D Landmark Detection and Tracking

Given an SI-SSM we are able to detect and track landmarks on 3D/4D sequences of range data. In order to perform the detection and tracking, we must first calculate the shape index values for the vertices of the input range mesh. This is done in the same manner as described in Section 2.2. Once we have these values calculated we can then apply the SI-SSM fitting algorithm to the input range mesh data.

First, an initialization phase is performed to give us a sufficient starting point to perform a local patch-based correlation search. During the initialization phase, to fit our model to the range data we learn the weight parameters *w* of the global shape by uniformly varying the weight vector to generate new instances of the SI-SSM. By performing this learning offline, for the initialization, we are able to have precise control over which shapes are constructed, ensuring that the new shapes constructed are valid (within the allowable shape domain). Iterative closest point (ICP) [3] is used to minimize the distance between each SI-SSM instance and the input range data. The patches from the instance of the SI-SSM with the lowest ICP matching score are used as the initialized starting landmarks for the SI-SSM. Given this global fit, we then calculate the local patch–based correlation score with the SI-SSM and the input range mesh. This correlation score is computed using a cross correlation template matching scheme [17]. The correlation score, CS_p, is computed for each patch as

$$CS_p = \frac{\sum_{i',j'}\left(P(i'j')\cdot R(i+i',j+j')\right)}{\sqrt{\sum_{i'j'}P(i'j')^2\cdot\sum_{i'j'}R(i+i',j+j')^2}} \tag{5}$$

where $P(i'j')$ is the computed shape index value at index (*i, j*) of the SI-SSM patch, and $R(i+i', j+j')$ is the summation between the shape index value at index *(i, j)* of the SI-SSM patch and the corresponding shape index value on the range mesh. The final correlation score, *CS*, is computed as

$$CS = \sum_{p=1}^{L} CS_p \tag{6}$$

This initial correlation score allows us to have a base line comparison for the local patch-based correlation search, as well as define tighter convergence criteria.

Once we have the initialized patches and initial correlation score we then perform a local search around each of the patches of the SI-SSM. For each patch in our model we construct a new patch of the same size around each of the $n \times n$ points of the original patch. For example, when *n=3*, we construct a patch

centered on each point of the original *3×3* patch, resulting in 9 new patches (as illustrated in Figure 3). The shape index values for each of these patches correspond to the shape index values of the vertices of the new patches. Using Equation 5 we compute a new CS_p for each of the new patches we created. The patch that gives us the highest correlation score is marked as the new patch of the SI-SSM. It is important to make sure that when all of the patches have been moved the new global shape of the face is with the allowable shape domain of ± 2 standard deviations from the mean. From Equation 4, we can derive the corresponding w_{SI} vector of the newly transformed SI-SSM by the following:

$$w_{SI} = {V_{SI}}^T (S_L - \bar{si}) \tag{7}$$

This new weight vector is constrained to be within the allowable shape domain, and we approximate a new shape by again utilizing Equation 4 with this weight vector.

Once we have the new approximated global shape of the face, iterative closest point is then used to again minimize the distance between the new SI-SSM instance and the range mesh. This process continues until convergence is reached. Convergence is defined by two main criteria:

(1) The computed correlation score, *CS*, for the transformed SI-SSM is higher than the computed score in the previous iteration (for the first iteration we make use of the correlation score computed in the initialization phase).
(2) The computed correlation score, *CS*, exhibits little to no change from the *CS* computed in the previous iteration.

If the first convergence criterion is satisfied after the first iteration after initialization, the transformed patches are discarded and the previously computed global patch shape is used. Due to this, we need to compute the initialization correlation score as it is possible in our initialization phase that our SI-SSM will find the best fit to the range mesh, and additional transformation(s) of the model are not required. Once we have the detected features for the current 4D mesh in the sequence, we then use ICP to move the landmarks to the next mesh in the sequence and continue the tracking of the sequence. The fitting process is then repeated with the previously detected landmarks used as the initial model fit. We are able to fit approximately 3 models per second with sufficient accuracy on an average PC configuration. Table 1 outlines the algorithm, Figure 3 shows an example illustration outlining the fitting process, and Figure 4 shows several sample 4D range models with detected patches using the SI-SSM algorithm.

SI-SSM FITTING ALGORITHM

Input: Range mesh model

1. Learn weight parameters for SI-SSM instances.
2. Initialize SI-SSM by using ICP to minimize distance between instances and input range mesh model.
3. Calculate correlation score, *CS*, for initialized SI-SSM.
4. Perform local patch-based correlation search.
5. Constrain transformed patches from step 4 to be within allowable shape domain.
6. Calculate new correlation score for newly transformed patches.
7. Compare new correlation score to score of previous iteration.
8. Repeat steps 4-7 until convergence.

Output: Detected patch landmarks on input range mesh.

Table 1. SI-SSM fitting algorithm.

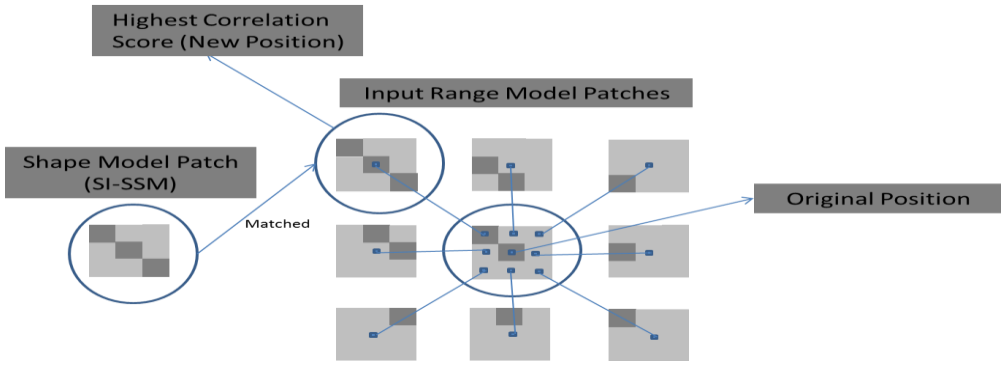

Fig. 3. Example of correlation search between a SI-SSM patch and input range model patch at size of $n \times n$, (where $n=3$ for instance).

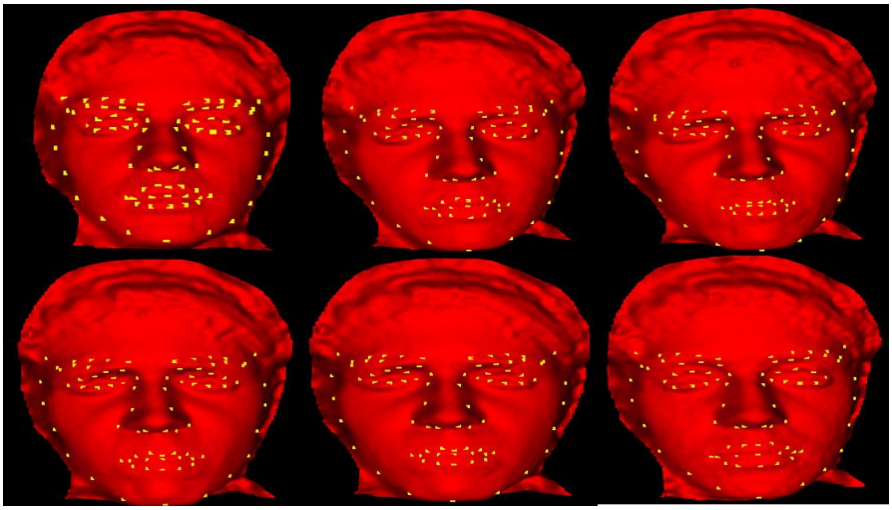

Fig. 4. Tracked frames from BU-4DFE displaying an angry expression.

4. Experiments and Evaluation

4.1 Databases

Five public face databases have been used for our study including three static and two dynamic databases (as shown in Table 2, and Figures 5 and 6 for examples).

(1) BU-3DFE [29] consists of 100 subjects each displaying one neutral expression and four intensity levels of six deliberate expressions.
(2) Eurecom Kinect Face Database [19] consists of 52 subjects, displaying 9 deliberate expressions, obtained through the Microsoft Kinect [16].
(3) FRGC 2.0 [22] consists of 466 subjects displaying two different deliberate expressions.
(4) BU-4DFE [30] consists of 101 subjects with sequences of six different deliberate expressions.
(5) BP4D-Spontaneous database [31] consists of 41 subjects, each consisting of 8 different spontaneous expression sequences (e.g., joy, embarrassment, surprise, disgust, fear, sadness, pain, and anger). The expressions were elicited through activities including film watching, interviews, and experiencing cold pressor test, etc. The database includes the 3D dynamic model sequences, texture videos, and annotated action units (AU). Table 2 lists more details pertaining to each database.

3D/4D DATABASE SUMMARIES						
Database	**Modality**	**Type**	**Number of Subjects**	**Resolution (# of vertices)**	**Number of Expressions**	**Number of Models**
3DFE	Static	Deliberate	100	8,000	7	2,500
4DFE	Dynamic	Deliberate	101	30,000	6	606 Sequences (100 frames/sequence)
FRGC 2.0	Static	Deliberate	466	100,000	2	932
BP4D	Dynamic	Spontaneous	41	50,000	8	328 Sequences (1,500 frames/sequence)
Eurecom	Static	Deliberate	52	65,000	9	936

Table 2. Summary of 3D/4D databases.

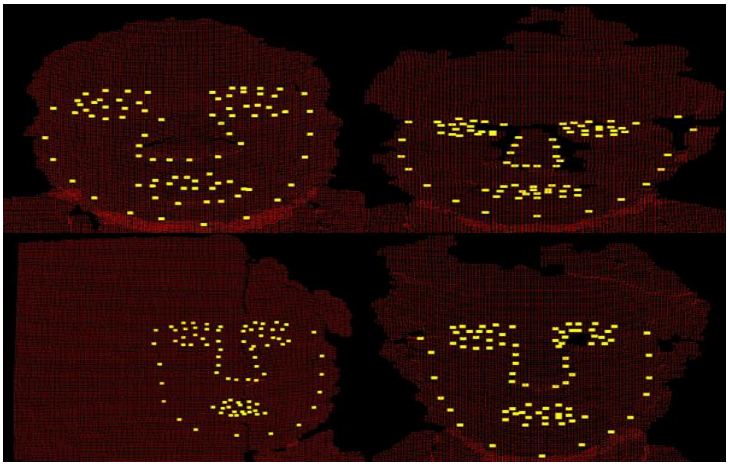

Fig. 5. Sample frames fit with SI-SSM algorithm from the Eurecom Kinect Face Database, showing robustness to occlusion, noise, and missing data.

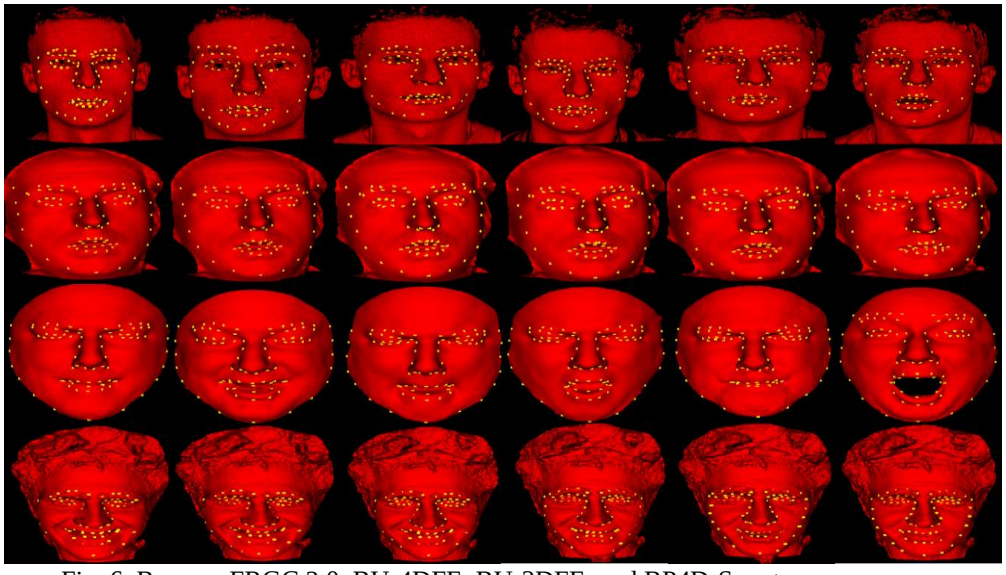

Fig. 6. By row: FRGC 2.0, BU-4DFE, BU-3DFE, and BP4D-Spontaneous.

4.2 Feature Detection and Tracking on Five Databases

To evaluate the accuracy of detecting and tracking landmarks using our SI-SSM method, we calculate the mean squared error between the ground truth and our detected/tracked landmarks (centroids of patches). We do this by calculating the one-point spacing between each landmark. The one-point spacing is defined as the closest pair of points on the 3D scans (0.5mm on the geometric surface). We treat the unit error as equal to 1 point spacing, so we can compute the average of the point distances between the sets. Table 3 details the error rates, mean squared error (MSE), for all five tested databases.

Database	**3DFE [29]**	**4DFE [30]**	**FRGC 2.0 [22]**	**BP4D [31]**	**Eurecom [19]**
Error Rate (MSE)	9.6	3.2	11.8	2.9	4.4

Table 3. Error rates for all 5 databases.

Note that the ground truth feature points that have been used for comparison in each database are obtained as follows:

(1) For 3DFE and 4DFE databases, we used the associated feature points (N=83) (released from the databases) as ground truth;
(2) For FRGC 2.0 and Eurecom databases, the ground truth feature points (N=83) were obtained through our manual annotation;
(3) For BP4D-Spontaneous database, the ground truth feature points (N=83) were obtained by a semi-automatic method: First, we utilized the Kinect face tracking API [16] and modified it for 3D range data tracking. To modify the Kinect face tracking algorithm a multi-rendering is done to render the 3D range data in a suitable depth and RGB format. Second, the 2D coordinates are converted into model space to acquire the 3D landmarks. Finally, we manually correct the feature points that were erroneously detected or mapped.

As can be seen from Table 3 our proposed algorithm performs well on the 4DFE and BP4D databases. The relatively higher error rate on the 3DFE can be attributed to the low resolution of this database. Also, the relatively higher error rates on the FRGC and Eurecom databases can be attributed to the greater level of noise and holes in these datasets.

4.3 Performance Evaluation

We also conducted three separate experiments on the BP4D database [31], which were split into the following categories: (1) expression segments, (2) rotations, and (3) occlusions/incomplete data. The details and errors statistics are detailed in the following sub-sections.

4.3.1 Spontaneous Expression Segments

The BP4D [31] includes 8 tasks that are meant to elicit an emotional response. Those emotions include happiness, sadness, surprise, embarrassment, fear, pain, anger, and disgust. We test the accuracy of our algorithm on segments containing 8 explicit expressions and plotted the average error in point spacings. For all of the tested expression segments there is a MSE of 3.1, the average point spacing error for each landmark (where L=83) along with the standard deviation can be seen in Figure 7. As can be seen from this figure, there is a small amount of variance between each of the 83 landmarks, with respect to their average error, thus showing it's robustness to different expressions. Figures 8 and 9 show two examples of spontaneous expressions.

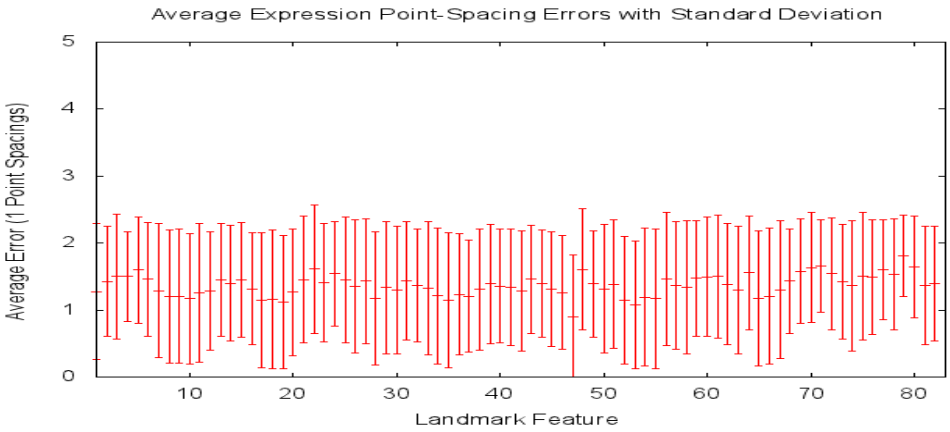

Fig. 7. Average error in point spacings of spontaneous expression sequences.

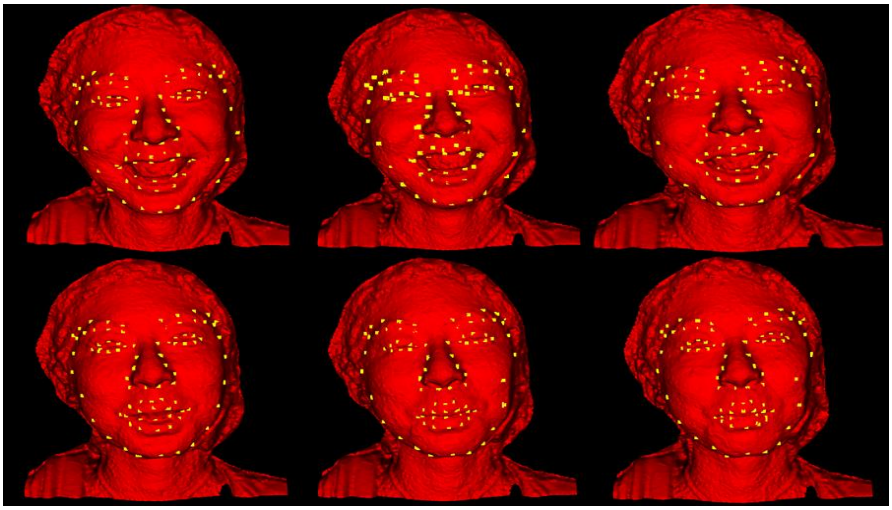
Fig. 8. Example of a tracked sequence of a subject in a joyful condition when watching a film.

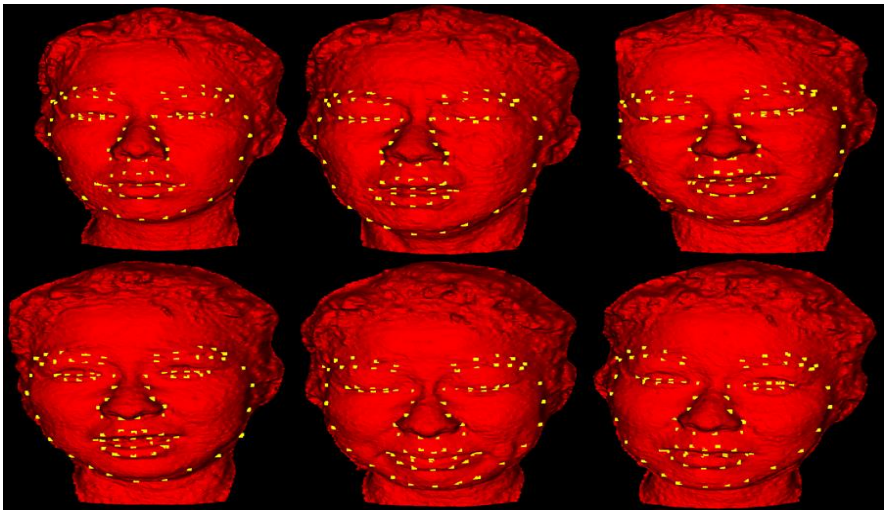
Fig. 9. Example of a tracked sequence showing a startled emotion.

4.3.2 Rotation Sequences

We also tested our algorithm on sequences that contain rotations only. We are able to successfully fit rotations in the range of [-90, 90]. Figure 12 shows an example of rotations between [0, 90]. For all tested rotation sequences there is a MSE of 3.2. Figure 10 shows the average point spacings error along

with the standard deviation for each of the, L=83, landmarks. As can be seen in Figure 10 the average error for rotations is fairly stable across all for all landmarks degrees, showing robustness to large rotations. Figure 11 show the average error, for all landmarks, across each of the degrees in the range [0, 90].

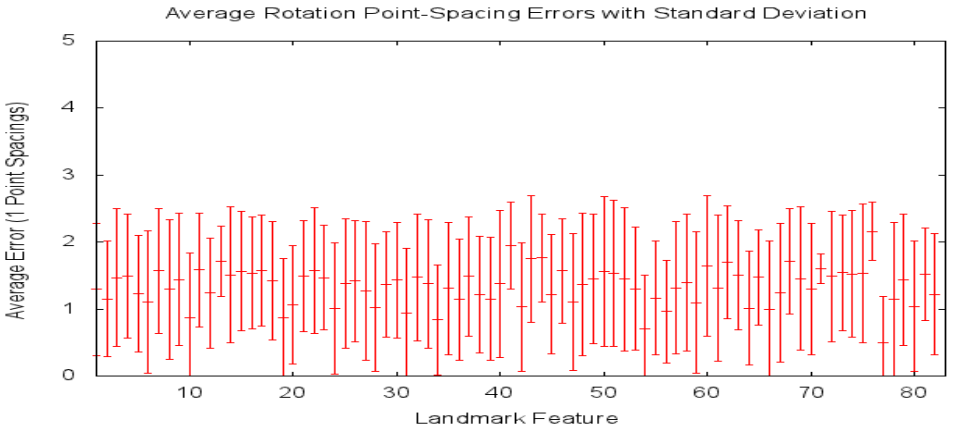

Fig. 10. Average error in point spacings for sequences displaying occlusions from rotations.

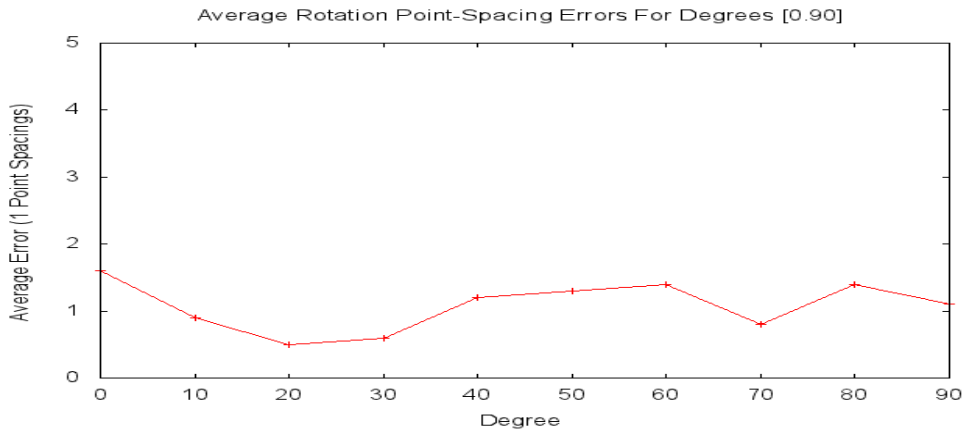

Fig.11. Average point spacing error in relation to rotation degree.

The results displayed in Figure 11 demonstrate the SI-SSM is robust to large rotations. For all rotations the average point spacing error is fairly consistent remaining under 2, with the largest error being 1.6 and the smallest error being 0.5. Figures 12 and 13 show two example sequential models with large head rotations.

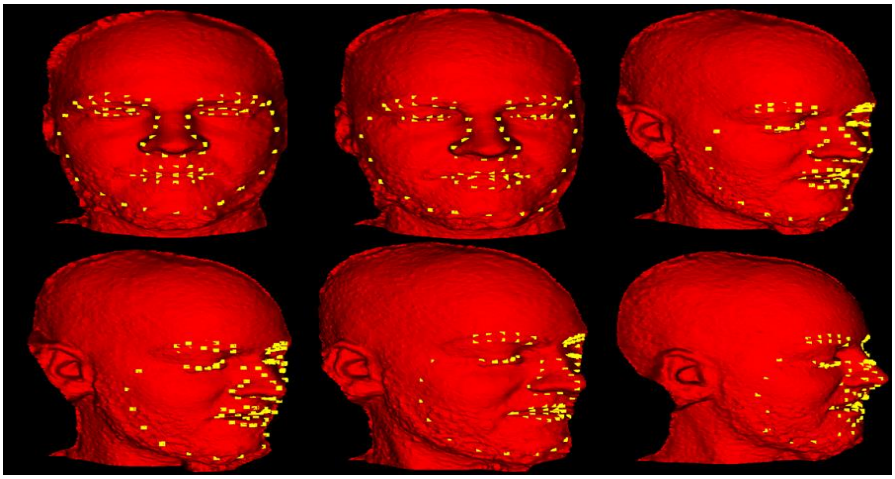

Fig. 12. Example of 4D data showing rotations from 0 degree to 90 degree

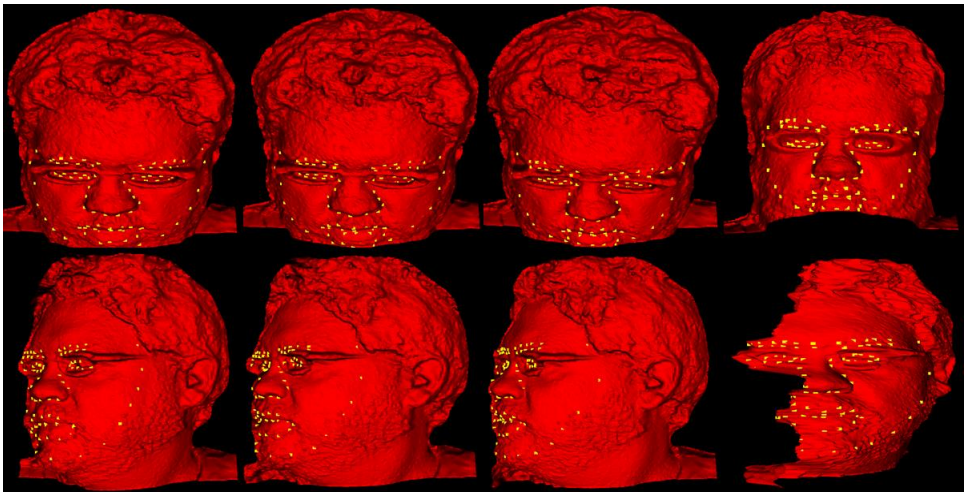

Fig. 13. Sample frames displaying pitch and yaw pose estimations. Top Row (Pitch): -20, -23, -27; Bottom Row (Yaw): -37, -49, -51;. *Note: The last column is the same model from the previous column. The models (with eye glasses) are rotated to the frontal view so that they show the mesh deformations (or missing pieces) that this degree of pose causes.*

4.3.3 Low Quality Sequences

We also tested the accuracy of our algorithm on sequences that contain low quality data. We define low quality data as missing data from self-occlusion (eye glasses, hand in front of face, etc.), noisy data (beards, distorted patches caused by 3D data capture, etc.), and incomplete scans containing holes and isolated patches. Figures 14-16 illustrate these types of low quality data sequences.

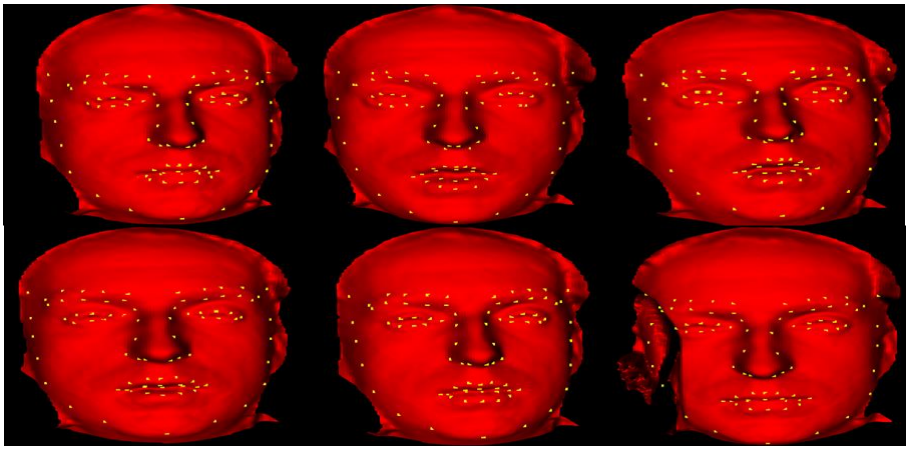

Fig. 14. Tracked frames displaying a surprised expression. NOTE: the bottom right frame in the sequence is missing data on the side of the face, and the SI-SSM still fits to the missing data showing robustness to missing data.

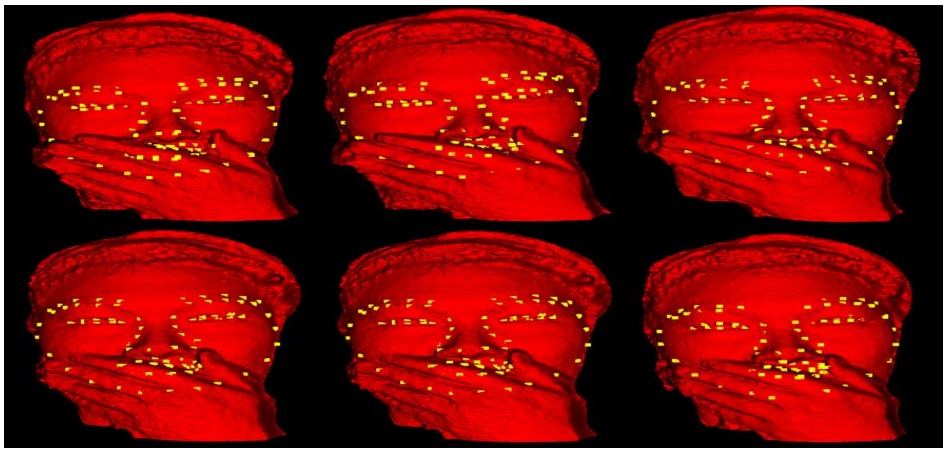

Fig. 15. Tracked data showing robustness to occlusion.

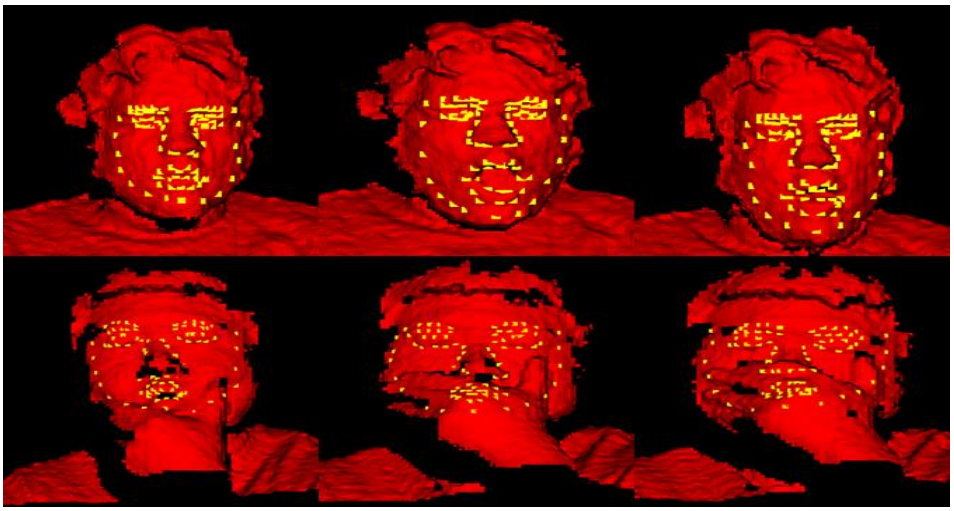

Fig. 16. Tracked data with partial occlusions showing robustness to noise and missing data.

Our test results on low quality data shows the MSE error is 3.6. Figure 17 shows the average error in point spacings, along with the standard deviation for low quality data While the MSE is slightly higher

and the average errors show more variance than other tested data, the SI-SSM is still able to successfully fit to this data with a generally low error rate, showing robustness to low quality data.

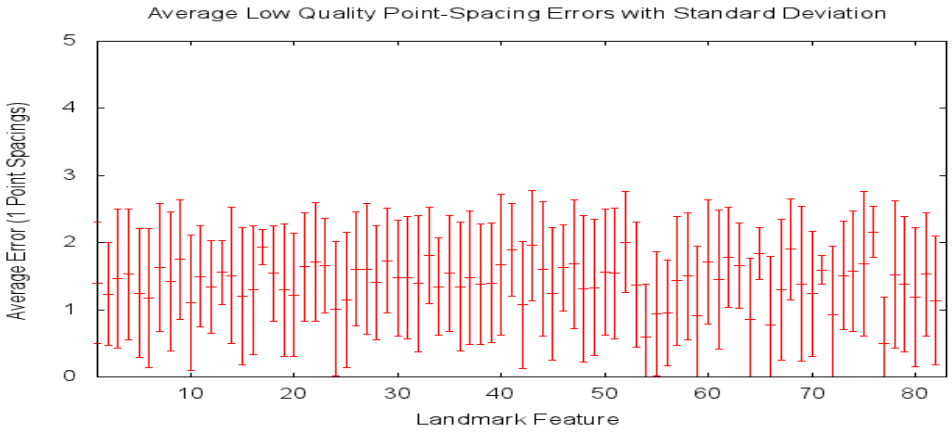

Fig. 17. Average error in point spacings of low quality data.\

4.4 Comparison to the State-of-the-Art

We also compared our SI-SSM algorithm, using 10% of the data for training and the rest for testing, against other state of the art algorithms. We compared our results against a 2D CLM mapped to 3D [24], TDSM [5], and Sun et al. [25] on the BU-4DFE and BP4D databases. For all comparisons we use the centroid landmark in each of the patches for comparisons, and report the MSE of the average point spacings. Note that the data tracked with the 2D CLM [24] only used 66 landmarks while we used 83 landmarks. In order to perform these experiments we selected the common sub-set of the two sets of landmarks, resulting in 49 landmarks for comparison. These landmarks comprise of the left and right eyes, nose, mouth and landmarks on the contour of the face. The 3D features mapped from 2D CLM have an error rate of 13.2 as compared to ours of 2.9. The high error rate of the 2D CLM based method can be attributed to frames where the tracking was lost and the method was unable to find a correct fit, as well as the mapping error from 2D to 3D. Figure 18 shows an example where the 2D CLM based method was unable to detect the correct landmarks while our SI-SSM was successful in detecting them. As can be seen from Table 4, which shows the results from these comparisons, our SI-SSM method outperforms the compared state of the art methods.

We also compare our work to Nair et al.[20] on the BU-3DFE database. For this experiment, we followed their detailed procedure. We selected four landmarks, the inner and outer corners of the left and right eyes, to compare to the ground truth. Figure 19 shows the mean normalized errors (MNE) for each of the four selected landmarks for all seven expressions in the database.

Comparison With State-of-the-Art				
	3D Mapped From 2D CLM [24]	TDSM [5]	Sun et al [25]	**SI-SSM**
BU-4DFE	N/A	3.7	6.3	**3.2**
BP4D	13.2	4.0	7.2	**2.9**

Table 4. Comparison of SI-SSM, TDSM, Sun et al, and 2D CLM Mapped to 3D. (MSE)

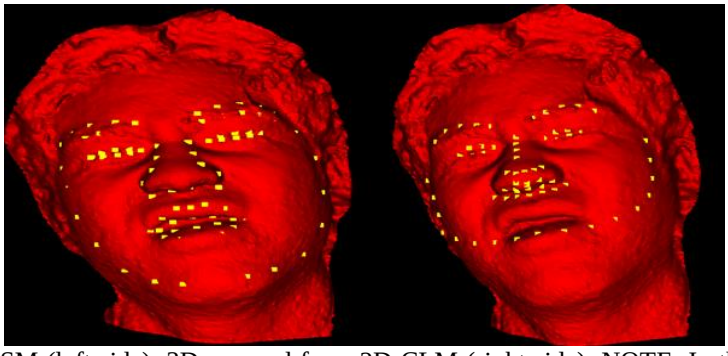

Fig. 18. SI-SSM (left side), 3D mapped from 2D CLM (right side). NOTE: In the sequence that contains this frame showing a rotated head pose with a painful expression from BP4D-Spontaneous database, there are approximately 100 frames that the 2D CLM fails to correctly fit (similar to this figure) whereas the SI-SSM is successful.

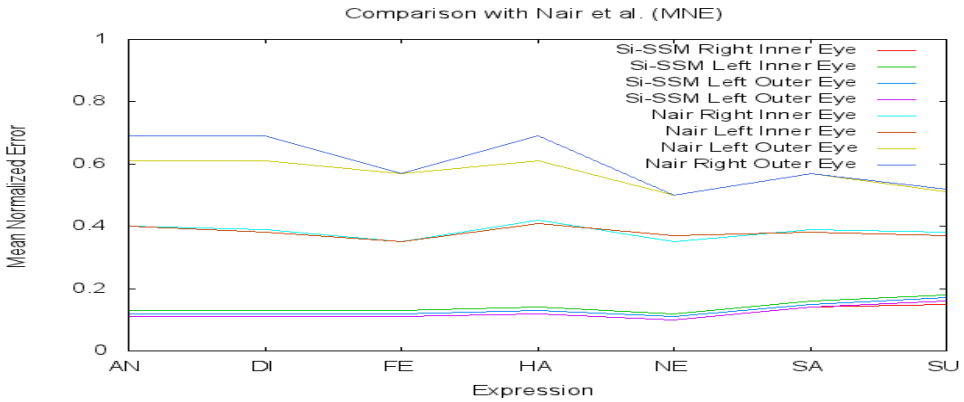

Fig. 19. MNE comparsion to Nair et al. [20] for expressions in BU-3DFE.

5. Applications

5.1 Posed and Spontaneous Facial Expression Classification

5.1.1. Approach

To validate our proposed method, we apply it to facial expression classification problems for both posed expressions and spontaneous expressions, respectively. We take the component based approach for the classification. Given the tracked feature points, we can easily segment the facial model into several component regions, such as the eyes, nose and mouth. Fig. 20(a) shows an example of the resulting segmentation.

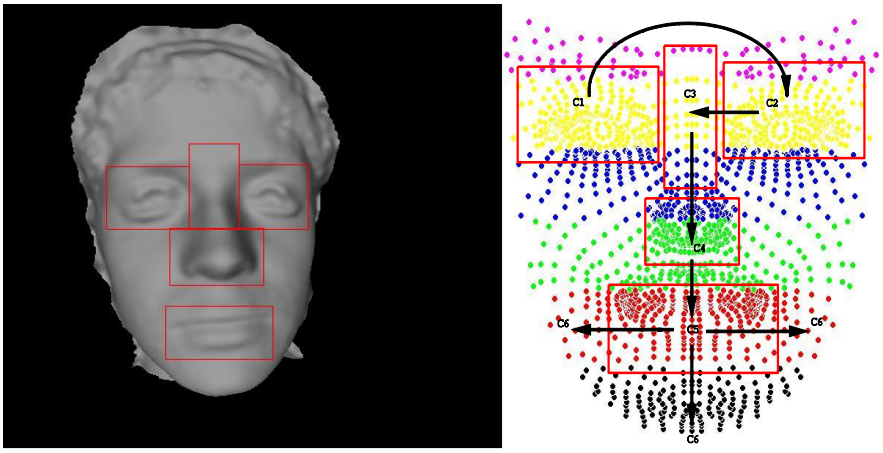

Fig.20. (a) Sample of component regions; (b) component-based HMM based on six component regions.

(i) 3D Component Feature Representation

3D facial models can be characterized by their surface primitive features. This spatial feature can be classified by eight types: convex peak, convex cylinder, convex saddle, minimal surface, concave saddle, concave cylinder, concave pit, and planar. Such a local shape descriptor provides a robust facial surface representation [12]. To label the model surface, we select the vertices of the component regions and then classify them into one of the primitive labels. The classification of surface vertices is based on the surface curvature computation [25]. After calculating the curvature values of each vertex, we use the categorization method [26] to label each vertex on the model. As a result, each range model is represented by a group of labels, that construct a feature vector: $G = (g_1, g_2,..., g_n)$, where g_i represents one of the primitive shape labels, and n equals the number of vertices in the component region.

Due to the high dimensionality of the feature vector G, where each of six component-regions contains between 300 and 700 vertices, we use a linear discriminant analysis (LDA) based method to reduce the feature space of each region. The LDA transformation maps the feature space into an optimal space where different subjects are easily differentiated. It then transforms the n-dimensional feature G to the d-dimensional optimized feature O_G ($d<n$).

(ii) Component based Spatial HMM Model

As shown in Figure 20(b), each frame of the 3D facial model is subdivided into six components (sub-regions) $C1$, $C2$, $C3$, $C4$, $C5$, and $C6$, including regions of the eyes, nose, nose bridge, mouth, and the remaining face. From $C1$ to $C6$, we construct a 1-D HMM [23] which consists of the six states ($N = 6$), corresponding to six regions.

We transform the labeled surface to the optimized feature space using the aforementioned LDA transformation. Given such an observation of each sub-region, we can train the HMM for each expression. Given a query face model sequence of length k, we compute the likelihood score for each frame, and use the Bayesian decision rule to decide which expression each frame is classified to. Since we obtain k results for k frames, we use a majority voting strategy to make a final decision. As such, the query model sequence is recognized as expression Y if Y is the majority result among k frames. This method tracks spatial dynamics of 3D facial surfaces, where the spatial components of a face give rise to the spatial HMM to infer the likelihood of each query model. Note that if k is equal to 1, the query model sequence becomes a single model for classification.

(iii) Component-based Temporal HMM Model

For the 3D expression sequences, we treat 6 frames as the 6 states of the HMM model for expression classification. When observing the state change of a local region across a sequence, we are able to use the facial features of the local region to train a temporal HMM. Each local region of a facial surface learns an HMM for each distinct expression separately. Given six local regions and *m* types of facial expressions, a total of $6 \times m$ T-HMMs are established for an entire facial surface ($m=6$ for BU-4DFE, and $m=8$ for BP4D-Spontaneous).

Since the features extracted from the six local regions could generate six different classification results, we use the majority voting strategy to determine the expression type of the subsequence. If more than two regions are classified as a same expression, such expression is taken as the recognized expression for this subsequence. If there is no majority expression to be recognized among the six regions ($R1$, ..., $R6$), the expression with the maximum likelihood (probability) of the region will be chosen as the recognized expression of this subsequence. This procedure is formulated as the following equation:

$$R_c = \underset{R_k}{\operatorname{argmax}}\left[\frac{P(\omega_{c*}^{k}|O^{R_k})}{\sum_{i=1}^{C} P(\omega_i|O^{R_k})}\right]_{k=1,2,\ldots,6} \tag{8}$$

where ω_{c*}^{k} is the expression type determined by the region R_k , ω_i is a trained HMM model, C is the number of trained HMM models, and O is an observation sequence. As a result, the expression of the region Rc with the maximum likelihood is selected as the recognized expression of the subsequence. In summary, the regional features of a facial surface are used to learn their temporal changes, and the classified expression is determined by either the majority voting or the maximum probability of observations of local regions.

(iv) Component-based Spatial-Temporal HMM Model

In order to determine a class of a certain expression, results from S-HMM and T-HMM are integrated as follows:

(a) The expression class follows one of the results of S-HMM and T-HMM if both are the same.
(b) The expression class follows the result of T-HMM if both are not the same, but the T-HMM has the more votes for a certain expression among six components than the votes of the other expressions from S-HMM.
(c) Vice versa, the expression class follows the result of S-HMM if both are not the same, but the S-HMM has the more votes for a certain expression among six frames than the votes of the other expressions from T-HMM.
(d) If none of above, the likelihoods (maximum probability) each individual expression from S-HMM and T-HMM are added, resulting six likelihoods (if six expressions). The one with highest likelihood is chosen as the recognized expression.

5.1.2 Experiment results on face expression classification using spatial-temporal HHM

The posed facial expression database (BU-4DFE) and spontaneous facial expression database (BP4D-Spontaneous) are used for experiments on face expression classification.

(i) Posed expressions: For training sequences of 4DFE, 1,200 sets of six consecutive frames were randomly chosen for training. Following the HMM training procedure ($k = 6$), we generated the spatial HMM and temporal HMM for each expression. The recognition procedure is then applied to classify the expression of each input sequence ($k = 6$). Based on the 10-fold cross

validation approach, the six prototypic facial expressions are classified with an accuracy of approximately 92.3%.

(ii) Spontaneous expressions: For training sequences of BP4D-Spontaneous, 2,560 sets of six consecutive frames were randomly chosen for training. Similar to the above procedure, we generated the spatial HMM and temporal HMM for each expression. The recognition procedure is then applied to classify the expression of each input sequence ($k = 6$). Based on the 10-fold cross validation approach, the eight spontaneous facial expressions are classified with an accuracy of approximately 83.7%.

5.2 Pose Estimation

Using the SI-SSM method, we are able to accurately detect landmarks on sequences consisting of rotations as seen in Figures 12 and 13. Using these detected landmarks, another natural application is for pose estimation. To do so, we use four fiducial points, which are a subset of the $N = 83$ landmarks we detected on the face models to construct a normal vector for pose representation. The four fiducial points used to calculate the normal vector are the left and right inner eye corners, as well as the left and right corners of the nose. Given these four points, a triangle is formed by each eye's inner corners and the average point of the two nose corners. The normal vector of such a triangle is relatively expression invariant, as can be seen in Fig. 21, thus representing the pose orientation of the head accordingly. We then use the relative rotation of the normal vector, compared to a model that is displaying a frontal view, to calculate the head pose angle.

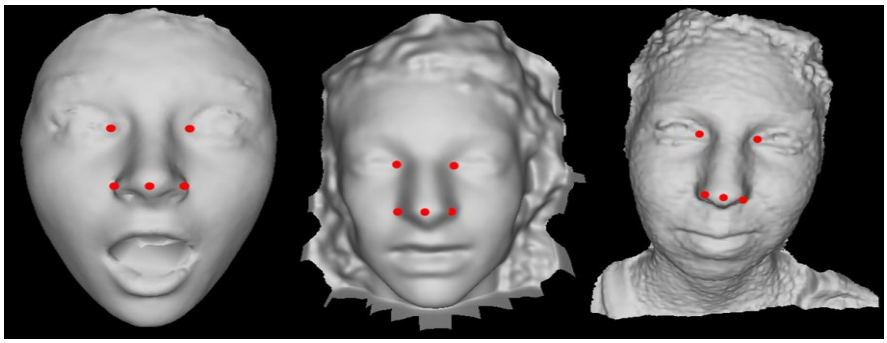

Figure 21. Sample illustrations, on BU-3DFE, BU-4DFE, and BP4D-Spontaneous, showing the landmarks used to create the normal vector used to determine head pose. *Note: the landmarks on the nose tip regions have been translated along the z axis for illustration purposes only. This landmark, being the average of the nose corners, would normally not be visible from this view.*

Once these steps are done we are able to compare the estimated poses with the ground truth. The comparisons show 2.53, 1.35, and 2.44 differences in degree across pitch, roll, and yaw respectively. Fig. 22 shows models displaying yaw, roll, and pitch with estimated poses.

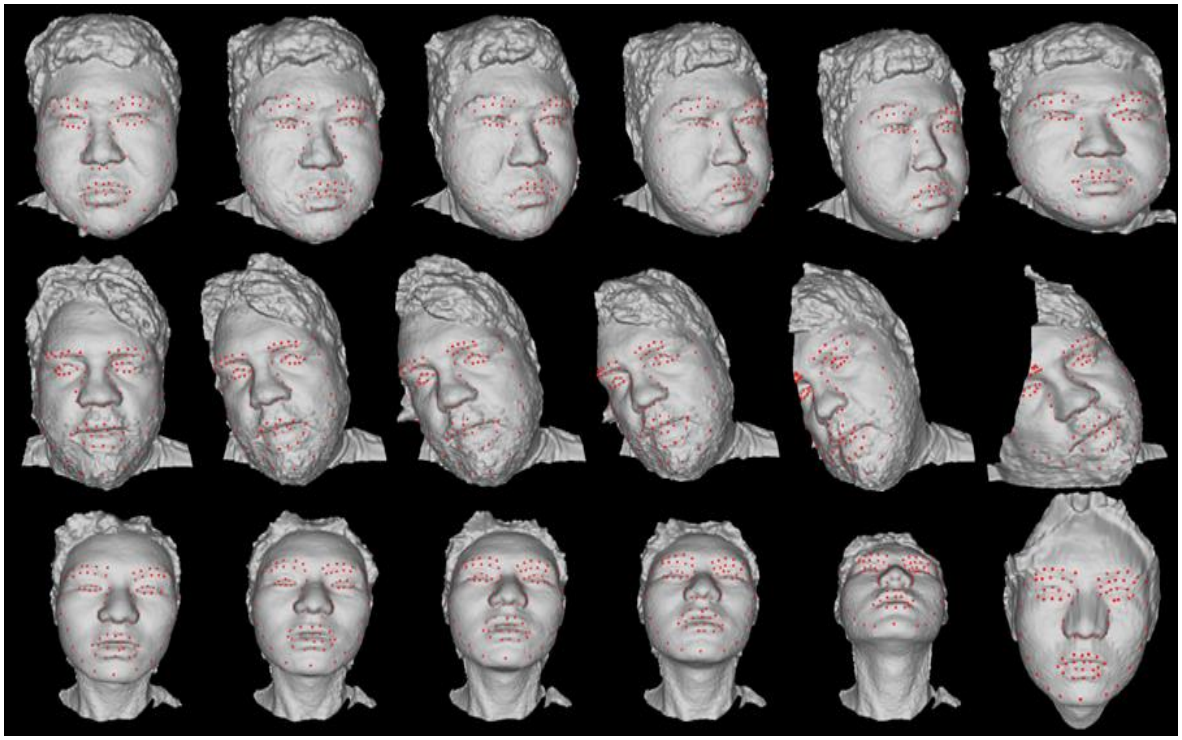

Fig. 22. Sample frames showing yaw (top), roll (middle), and pitch (bottom) pose variations with estimated poses: yaw (0, 10, 20, 30, 40), roll (0, 10, 20, 30, 40), and pitch (0, 10, 20, 30, 50). Note: The last column in each row is the same model from the previous column. This frontal view is to show how the pose change could cause mesh missing or distorted.

6. Conclusion and Future Work

In this paper we have presented a novel method of detecting and tracking landmarks on 3D and 4D data using a shape index-based statistical shape model. The SI-SSM has been tested on five public 3D/4D face databases. The SI-SSM has shown robustness to rotations, occlusions, and low quality data, as well as superiority to the compared state of the art methods, given only the geometric information used.

In our future work we will investigate what effect the patch size has on the fitting results, as well as using machine learning techniques to learn the best match between the input mesh patches and the SI-SSM patches. We will also investigate other optimal geometric features with a combination of texture information for applications in real-time face expression analysis. In addition, this approach could be generalized to apply for the feature detection and tracking on the other non-facial data, for instance, the human gesture (hand and body) feature detection and tracking in the future.

7. Acknowledgement

This material is based upon work supported in part by the National Science Foundation under grants IIS-1051103, CNS-1205664, and IIS-0541044.

References

[1] T. Baltrusaitis, P.Robinson, and L. Morency, “3D constrained local model for rigid and non-rigid facial tracking,” CVPR 2012.

[2] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar, “Localizing Parts of Faces Using a Consensus of Exemplars,” IEEE Trans. PAMI, (35)12:2930-2940, 2013.

[3] P. Besl, N. McKay, "A method of registration of 3D shapes," IEEE Trans. on PAMI vol. 14, pp. 239-256, 1992.
[4] V. Blanz, T. Vetter, "A morphable model for the synthesis of human faces," SIGGRAPH, 1999.
[5] S. Canavan, X. Zhang, L. Yin, "Fitting and tracking 3D/4D facial data using a temporal deformable shape model," International Conference on Multimedia & Expo (ICME), 2013.
[6] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," Intl. Journal of Computer Vision, 107:177-190, 2014.
[7] T. Cootes, C. Taylor, D. Cooper, J. Graham, "Active shape models-their training and application," CVIU vol. 61 pp. 18-23, 1995.
[8] T.F.Cootes, K. Walker, and C.J. Taylor, "View-based active appearance models," Image and Vision Computing 20(9): 657-664, 2002
[9] D. Cristinacce and T. Cootes, "Feature Detection and Tracking with Constrained Local Models," BMVC, 2(5), 2006.
[10]C.Dorai and A. Jain, "Cosmos – A Representation Scheme for 3D Free-form Objects," IEEE Trans. PAMI, (19)10, 1997.
[11]G. Fanelli, M. Dantone, and L.V. Gool, "Real time 3D face alignment with random forests-based active appearance models," FGR, 2013.
[12]P.Flynn and A. Jain, "Surface classification: Hypothesis testing and parameter estimation," IEEE CVPR 1988.
[13]L. Gu and T. Kanade, "A generative shape regularization model for robust face alignment," ECCV, 2008.
[14]P.Guan, Y.Yu, and L. Zhang, "A novel facial feature point localization method on 3D faces," International Conference on Image Processing 2007.
[15]L. Jeni, A. Lorinca, T. Nagy, Z. Plotai, J. Sebok, Z. Szabo, D. Takacs, "3D shape estimation in video sequences provides high precision evaluation of facial expressions, "Image and Vision Computing, (30)10: 785-795, 2012.
[16]Kinect for Windows. Microsoft Corporation, Redmond WA.
[17]J. Lewis, "Fast Template Matching," Vision Interface. 1995.
[18]I. Matthews and S. Baker, "Active Appearance Models Revisited," International Journal Computer Vision, (60):135-164, 204.
[19]R. Min, N. Kose, and J Dugelay, "KinectFaceDB: A Kinect Database for Face Recognition." IEEE Trans. on SMC, 44(11): 1534-1548, 2014.
[20]P. Nair, A. Cavallaro, "3D face detection landmark localization, and registration using a point distribution model," IEEE Trans. on Multimedia (11), 2009.
[21]P. Perakis, G. Passalis, T. Theoharis, and I.A. Kakadiaris, "3D Facial Landmark Detection Under Large Yaw and Expression Variations," IEEE Trans. on PAMI, 35(7):1552-1564, 2013.
[22]P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, J. Worek, "Overview of the face recognition grand challenge," IEEE CVPR 2005.
[23]Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition Proceedings of IEEE, 77(2), 1989.
[24]J.M. Saragih, S. Lucey, J.F. Cohn, "Deformable model fitting by regularized landmark mean-shift," Intl. Journal Computer Vision, 91(2), 2011.
[25]Y. Sun, X. Chen, M. Rosato, L. Yin, "Tracking vertex flow and model adaption for 3D spatio-temporal face analysis," IEEE Trans. on SMC A vol. 40 2010.
[26]J. Wang, L. Yin, X. Wei, Y. Sun, "3D facial expression recognition based on primitive surface feature distribution," CVPR 2006.
[27]T. Weise, S. Busaziz, H. Li, and M. Pauly, "Real-time performance-based facial animation," ACM Trans. Graphics 2011.
[28]X. Xiong and F. De la Torre, "Surpervised Descent Method and its Applications to Face Alignment," CVPR 2013.
[29]L. Yin, X. Wei, Y. Sun, J. Wang, M. Rosato, M., "A 3D facial expression database for facial behavior research," FGR 2006.

[30]L. Yin, X. Chen, Y. Sun, et al., "A high-res 3D dynamic facial expression database," FGR 2008.
[31]X. Zhang, L. Yin, J. Cohn S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. Girard, "BP4D-Spontaneous: A high resolution 3D dynamic facial expression database," Image and Vision Computing, 10, 2014.
[32]X. Zhao et al, "Accurate landmarking of three-dimensional facial data in the presence of facial expressions and occlusions using a three-dimensional statistical facial feature model," IEEE Trans. SMC Part B, 41(5): 1417-1428, 2011.
[33]D. Zhou, D. Petrovska-Delacretaz, and B. Dorizzi, "3D active shape model for automatic facial landmark location trained with automatically generated landmark points," ICPR 2010.
[34]I. A. Kakadiaris, G. Passalis, G. Toderick, M. N. Murtuza, Y. Lu, N. Karampatzikas, T. Theohari, "Three-dimensional face rec. in the presence of facial expressions: An annotated deformable model approach," IEEE Trans. on PAMI 29(4): 640-649, 2007.
[35]M. Segundo, C. Queirolo, O.R.P. Bellon, L.Silva, "Automatic 3D facial segmentation and landmark detection," International Conference on Image Analysis and Processing 2007.
[36]T. Fang, X. Zhao, O. Ocequeda, S. Shan, and I. Kakadiaris, 3D/4D facial expression analysis: an advanced annotated face model approach, Image and Vision Computing, 30(10):738-749, 2012
[37]T Fang, X Zhao, O Ocegueda, SK Shah, IA Kakadiaris, 3D facial expression recognition: A perspective on promises and challenges, IEEE Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011.
[38]H. Li, D. Huang, JM. Morvan, Y. Wang, and L. Chen, Towards 3D Face Recognition in the Real: A Registration-Free Approach using Fine-Grained Matching of 3D Keypoint, International Journal of Computer Vision, pp. 1-14, 2014.
[39]J. Sun, D. Huang, Y. Wang, and L. Chen, A Coarse-to-Fine Approach to Robust 3D Facial Landmarking via Curvature Analysis and Active Normal Model, IEEE International Joint Conference on Biometrics (IJCB), 2014.
[40]G. Sandbach, S. Zafeiriou, M. Pantic, L. Yin, Static and dynamic 3D facial expression recognition: A comprehensive survey, Image and Vision Computing 30 (10) (2012) 683-697.
[41]H. Soyel, H. Demirel, Facial expression recognition using 3D facial feature distances, Image Analysis and Recognition (2007) p831-838.
[42]J. Wang, L. Yin, X. Wei, Y. Sun, "3D facial expression recognition based on primitive surface feature distribution," CVPR 2006.
[43]H. Tang, T. S. Huang, 3d facial expression recognition based on properties of line segments connecting facial feature points, in: 8th IEEE international conference on automatic face and gesture recognition and workshops (FG), 2008.
[44]S. Berretti, A. D. Bimbo, P. Pala, B. B. Amor, M. Daoudi, A set of selected sift features for 3D facial expression recognition, International Conference on Pattern Recognition, 2010, pp. 4125-4128.
[45] E. Ong, and R. Bowden, "Robust facial feature Tracking using shape-constrained multiresolution-selected linear predictors," IEEE Trans. on PAMI, 33(9):1844-1859, 2011.
[46]P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, J. Worek, "Overview of the face recognition grand challenge," IEEE CVPR 2005.
[47]P. Szeptycki, M. Ardabilian, and L. Chen, "A coarse-to-fine curvature analysis-based rotation invariant 3D face Landmarking," Biometrics: Theory, Applications and Systems 2009.
[48]H. Dibeklioglu, A.A.Salah, and L. Akarun, "3D facial landmarking under expression, pose, and occlusion variations," Biometrics: Theory, Applications and Systems 2008.
[49]A. Salazar, S. Wuhrer, C. Shu, and F. Prieto, "Fully automatic expression-invariant face correspondence," Machine Vision and Applications, 25:859-879, 2014
[50]U. Bonde, V. Badrinarayanan, and R. Cipolla, "Multi Scale Shape Index for 3D Object Recognition," Scale Space and Variational Method in Computer Vision, 2013.
[51]H. Chen, and B. Bhanu, "3D free-form object recognition in range images using local surface patches," Pattern Recognition Letters, (28), pp. 1252-1262, 2007.
[52]Y, Wang, B. Peterson, and L. Staib, "Shape-Based 3D Surface Correspondence Using Geodesics and Local Geometry," Computer Vision and Pattern Recognition, 2000.

[53]B. Amberg, R. Knothe, and T. Vetter, “Expression Invariant 3D Face Recognition with a Morphable Model,” Face and Gesture Recognition, 2008.