Sparse and Dense 3D Face Modelling based on 2.5D AAM Approach from a Single Image

Amir Jamali, Dr. A.Raie

Abstract

In this thesis, an algorithm is developed which, for a given 2D frontal face input image, generates a 3D model of the face in the form of a point cloud and subsequently derives the face depth image from this model. Besides, if the camera calibration information is given, the algorithm could provide accurate rotation and translation of the face concerning the camera; otherwise, relative rotation and translation of one face image to another is obtainable. To do so, it starts by driving a deformable shape model from the 3D face landmarks (83 points) and a deformable appearance model from 2D face images in BU3DFE dataset. Then, by altering shape, appearance and pose parameters iteratively, it fits the deformable model to a 2D face input image and obtains its optimal model. In fitting process, the idea in 2.5D AAM algorithm with some modifications is used. In the modified version which is one of this thesis novelties, a cost function is defined, and initial shape parameters are estimated, resulting enhancement of AAM performance for unseen data, much fewer iterations than former approaches and robustness to added distortions to the initial shape parameters. From the output of the fitting process, which is a sparse 3D point cloud and via subdividing and 3D affine transform, the depth image is extracted. In this regard, to project 3D points to 2D image plane, the proposed approach uses perspective projection in case of knowing camera calibration information or, otherwise uses a weak perspective model with changing scale, which is another novelty of this thesis. Different properties of the proposed algorithm are evaluated. It is shown that the proposed algorithm is faster, since it converges with less than eight iterations, compared to 20 for other approaches. Its robustness to added distortions to the initial shape parameters, with any standard deviation, in the fitting process is shown. The mean and the standard deviation of the depth image error for the proposed algorithm and the GEM approach are compared. It is verified that the proposed algorithm has less reconstruction error.

Brief Review to Thesis

First of all, a Deformable 3D landmark and Appearance image (mean appearance) is built from the dataset which can be deformed by **p** and $\boldsymbol{\lambda}$ parameters respectively. (fig1)

After then proposed method tries to estimate **p**(3D landmark or shape parameter), **q**(pose parameter), $\boldsymbol{\lambda}$(appearance parameter) parameters minimizing following cost function in order to fit the model to the face image:

$$argmin_{pq\lambda} \sum_{x_p \in S_0 p} [A_0(x_p) + \sum_{i=1}^{m} \lambda_i A_i(x_p) - I(W(x_p, p, q))]^2$$

In above cost function A_0 is mean appearance and A_i is ith of m most significant eigen vector of appearance. $I(W(x_p, p, q))$ is projected warped image of input face image depending on shape and pose parameters. Block diagram of fitting process is shown in fig3. After fitting process termination, 2D & 3D landmark is generated for the face image. Fitting process for two sample face image is shown in fig2.

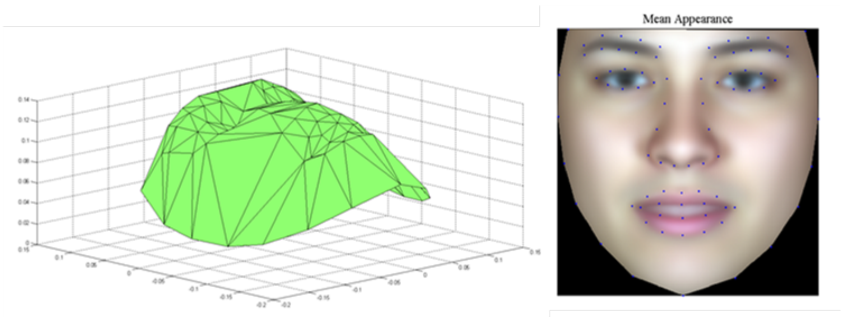
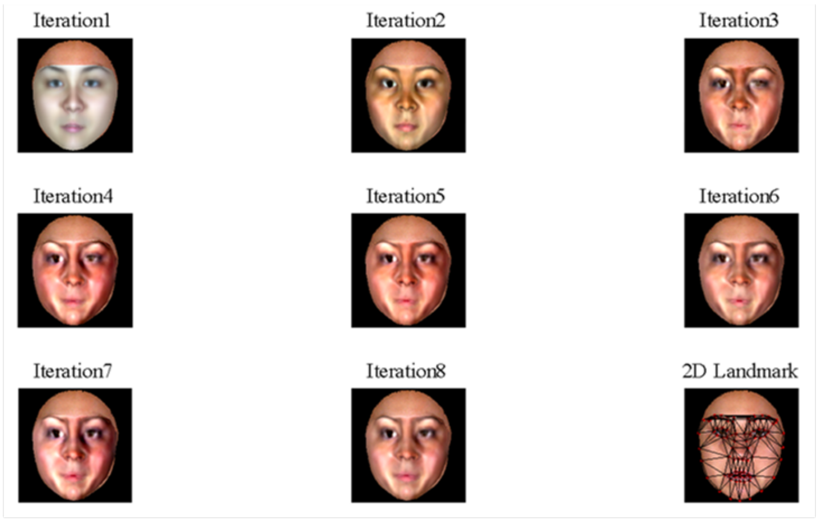


Fig1 mean 3D landmark & mean appearance



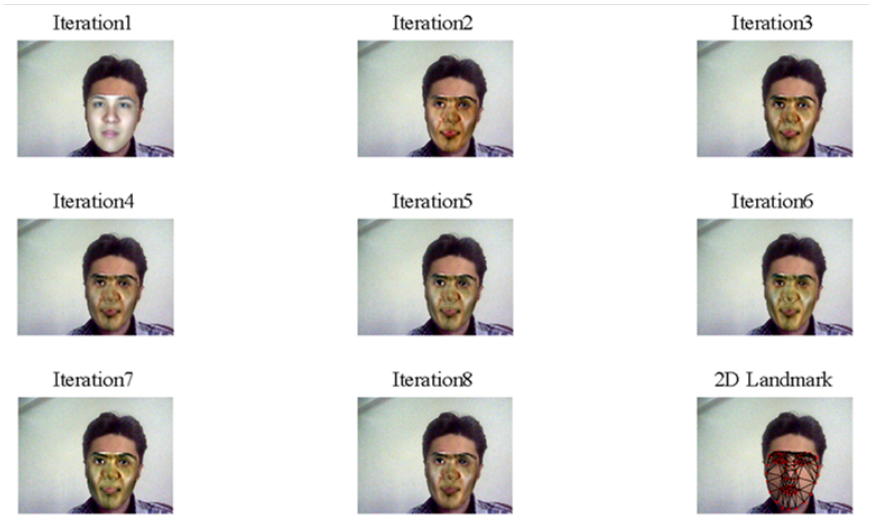


Fig2 fitting process for two sample face image

By knowing camera calibration data(intrinsic parameters) our proposed method calculates relative rotation and translation of face to the camera in addition to 2D & 3D landmark results. Block diagram of the fitting process by knowing camera calibration parameters is shown in fig4.

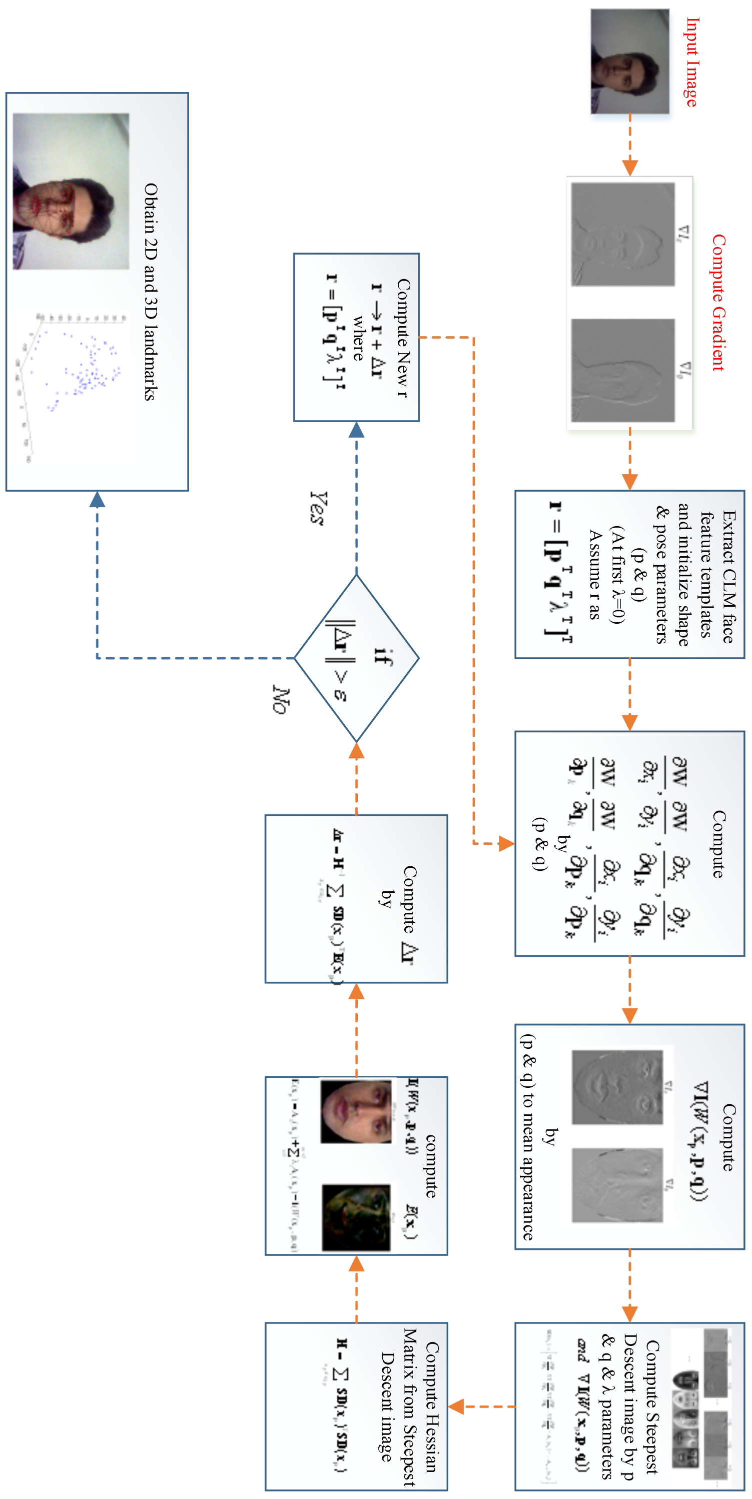
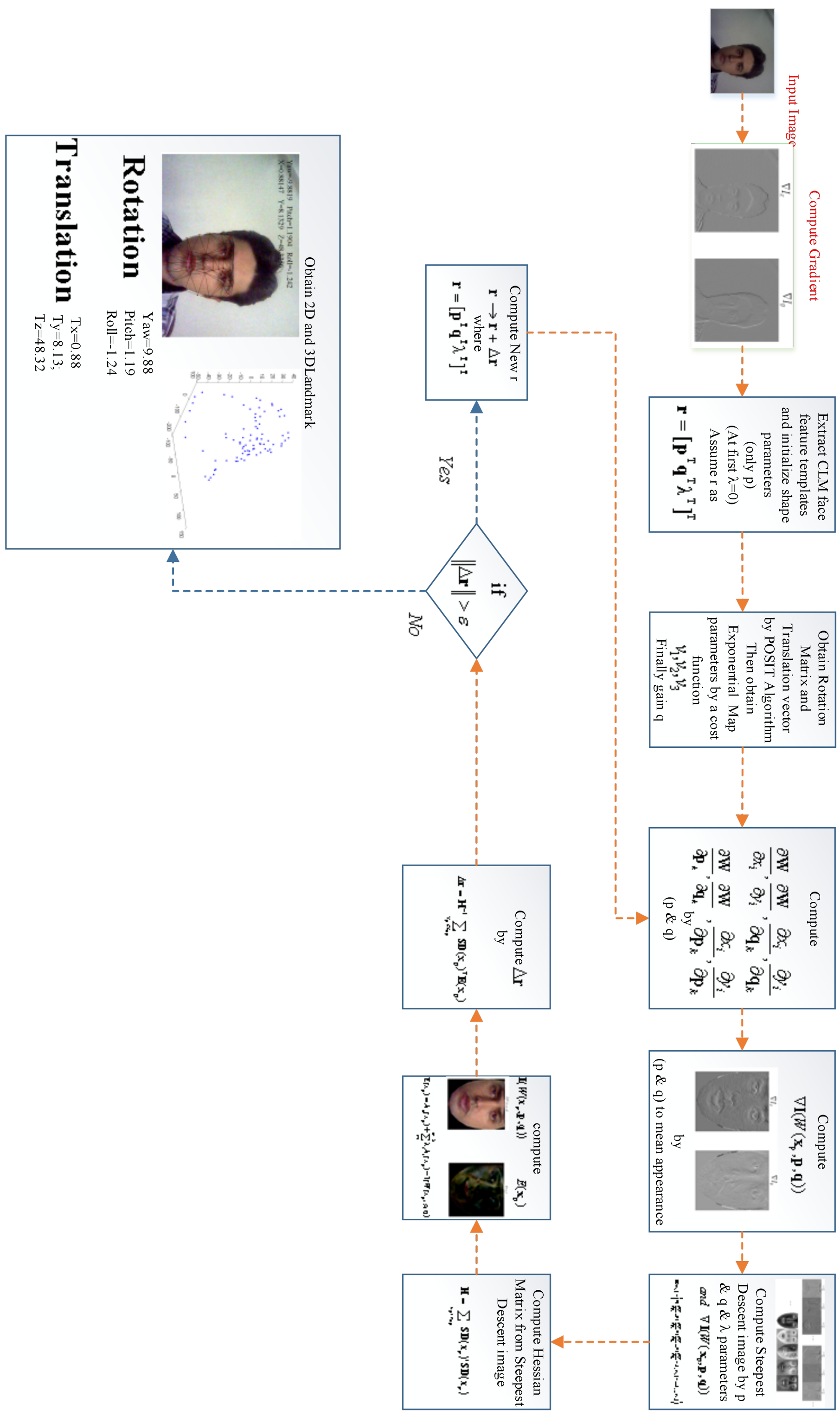

Input Image
Compute Gradient
∇I_x
∇I_y
Extract CLM face feature templates and initialize shape & pose parameters (p & q) (At first λ=0) Assume r as
$\mathbf{r} = [\mathbf{p}^T \mathbf{q}^T \lambda^T]^T$
Compute $\frac{\partial W}{\partial x_i}, \frac{\partial W}{\partial y_i}, \frac{\partial x_i}{\partial \mathbf{q}_k}, \frac{\partial y_i}{\partial \mathbf{q}_k}, \frac{\partial W}{\partial \mathbf{q}_k}, \frac{\partial W}{\partial \mathbf{p}_k}$ by $\frac{\partial x_i}{\partial \mathbf{p}_k}, \frac{\partial y_i}{\partial \mathbf{p}_k}$ (p & q)
Compute $\nabla I(W(\mathbf{x}_i, \mathbf{p}, \mathbf{q}))$ by (p & q) to mean appearance
Compute Steepest Descent image by p & q & λ parameters and $\nabla I(W(\mathbf{x}, \mathbf{p}, \mathbf{q}))$
Compute Hessian Matrix from Steepest Descent image
$\mathbf{H} = \sum \mathbf{SD}(\mathbf{x}_i)^T \mathbf{SD}(\mathbf{x}_i)$
compute $E(\mathbf{x}_i)$
$I(W(\mathbf{x}, \mathbf{p}, \mathbf{q}))$
Compute $\Delta \mathbf{r}$ by
$\Delta \mathbf{r} = \mathbf{H}^{-1} \sum \mathbf{SD}(\mathbf{x}_i)^T E(\mathbf{x}_i)$
if $\|\Delta \mathbf{r}\| > \varepsilon$
Yes
No
Compute New r
$\mathbf{r} \to \mathbf{r} + \Delta \mathbf{r}$
where
$\mathbf{r} = [\mathbf{p}^T \mathbf{q}^T \lambda^T]^T$
Obtain 2D and 3D landmarks

Fig3 Block diagram of fitting process



Input Image
Compute Gradient
∇I_x
∇I_y
Extract CLM face feature templates and initialize shape parameters (only p) (At first λ=0) Assume r as
$\mathbf{r} = [\mathbf{p}^T \mathbf{q}^T \lambda^T]^T$
Obtain Rotation Matrix and Translation vector by POSIT Algorithm Then obtain Exponential Map parameters by a cost function $\gamma_1, \gamma_2, \gamma_3$ Finally gain q
Compute $\frac{\partial W}{\partial x_i}, \frac{\partial W}{\partial y_i}, \frac{\partial x_i}{\partial q_k}, \frac{\partial y_i}{\partial q_k}, \frac{\partial W}{\partial p_i}, \frac{\partial W}{\partial q_k}, \frac{\partial x_i}{\partial p_k}, \frac{\partial y_i}{\partial p_k}$ by (p & q)
Compute $\nabla I(W(\mathbf{x}_i, \mathbf{p}, \mathbf{q}))$ by (p & q) to mean appearance
Compute Steepest Descent image by p & q & λ parameters and $\nabla I(W(\mathbf{x}_n, \mathbf{p}, \mathbf{q}))$
Compute Hessian Matrix from Steepest Descent image
$\mathbf{H} = \sum SD(\mathbf{x}_n)^T SD(\mathbf{x}_n)$
compute
$I(W(\mathbf{x}_n, \mathbf{p}, \mathbf{q}))$
$E(\mathbf{x}_n)$
Compute Δr by
$\Delta\mathbf{r} = \mathbf{H}^{-1} \sum SD(\mathbf{x}_n)^T E(\mathbf{x}_n)$
if $\|\Delta \mathbf{r}\| > \varepsilon$
Yes
No
Compute New r
$\mathbf{r} \rightarrow \mathbf{r} + \Delta\mathbf{r}$
where
$\mathbf{r} = [\mathbf{p}^T \mathbf{q}^T \lambda^T]^T$
Obtain 2D and 3DLandmark
Rotation
Translation
Yaw=9.88
Pitch=1.19
Roll=-1.24
Tx=0.88
Ty=8.13;
Tz=48.32

Fig4 Block diagram of fitting process by knowing Camera Calibration Parameters

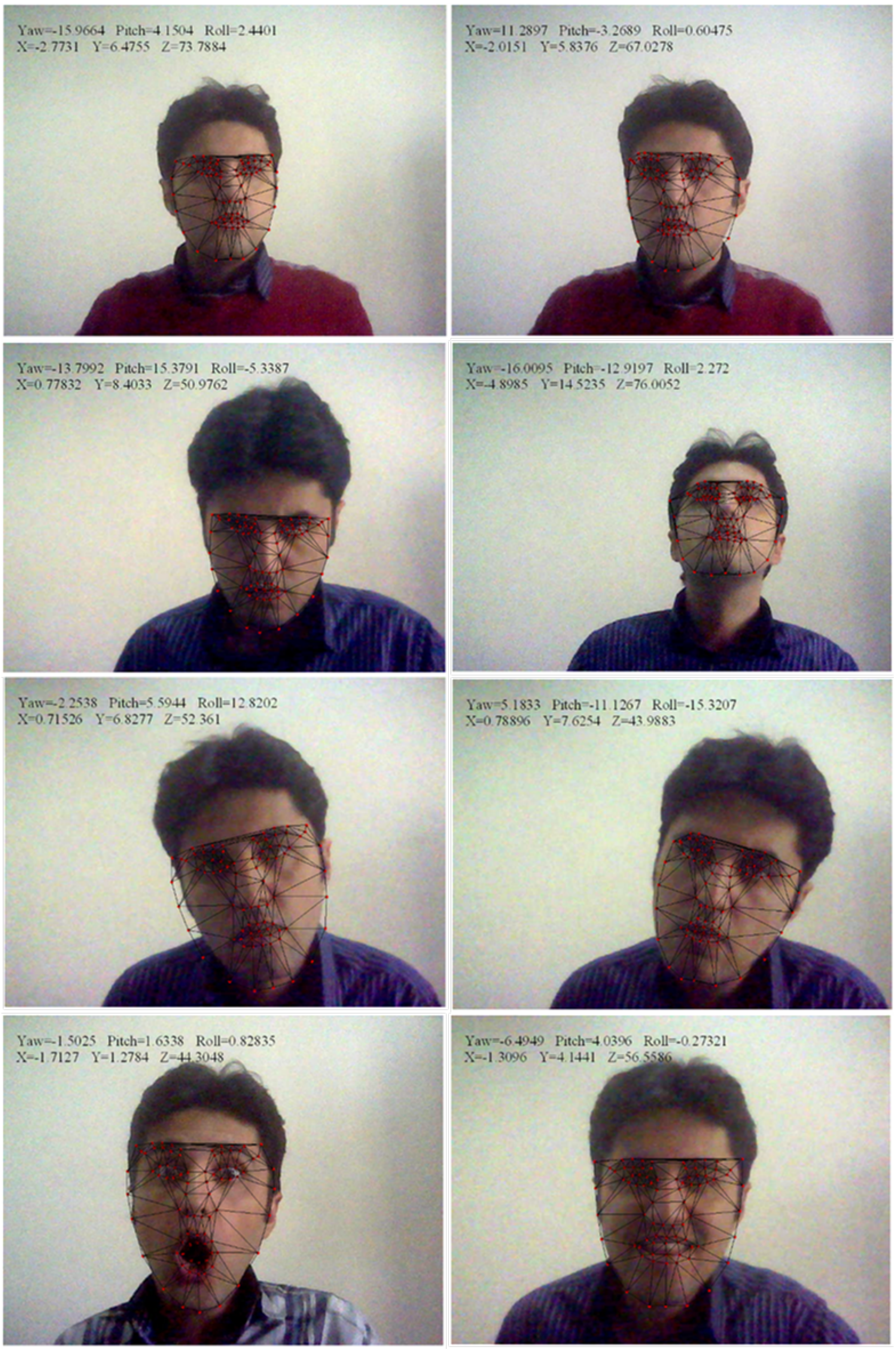

Yaw=-15.9664 Pitch=4.1504 Roll=2.4401
X=-2.7731 Y=6.4755 Z=73.7884
Yaw=11.2897 Pitch=-3.2689 Roll=0.60475
X=-2.0151 Y=5.8376 Z=67.0278
Yaw=-13.7992 Pitch=15.3791 Roll=-5.3387
X=0.77832 Y=8.4033 Z=50.9762
Yaw=-16.0095 Pitch=-12.9197 Roll=2.272
X=-4.8985 Y=14.5235 Z=76.0052
Yaw=-2.2538 Pitch=5.5944 Roll=12.8202
X=0.71526 Y=6.8277 Z=52.361
Yaw=5.1833 Pitch=-11.1267 Roll=-15.3207
X=0.78896 Y=7.6254 Z=43.9883
Yaw=-1.5025 Pitch=1.6338 Roll=0.82835
X=-1.7127 Y=1.2784 Z=44.3048
Yaw=-6.4949 Pitch=4.0396 Roll=-0.27321
X=-1.3096 Y=4.1441 Z=56.5586

Fig5 Some fitting process results by knowing intrinsic camera parameters

Some Fitting process results by knowing intrinsic camera parameters is shown in fig5; as you can see there, after fitting process, relative rotation and translation of face to the camera is given, plus 2D *&* 3D landmark. In next step our proposed approach generates a dense point cloud of face and face depth image consequently from sparse 3D landmark obtained in the previous step; let see how we could do this. In order to do that at first, a mean depth image is constructed by applying [1]GPA to all dataset's depth image via sparse 3D landmarks (fig6). Now we have mean depth with a mean sparse 3D landmark on one side and sparse 3D landmark obtained from previous step with no depth on another side. So we want to generate Depth image on the second side.

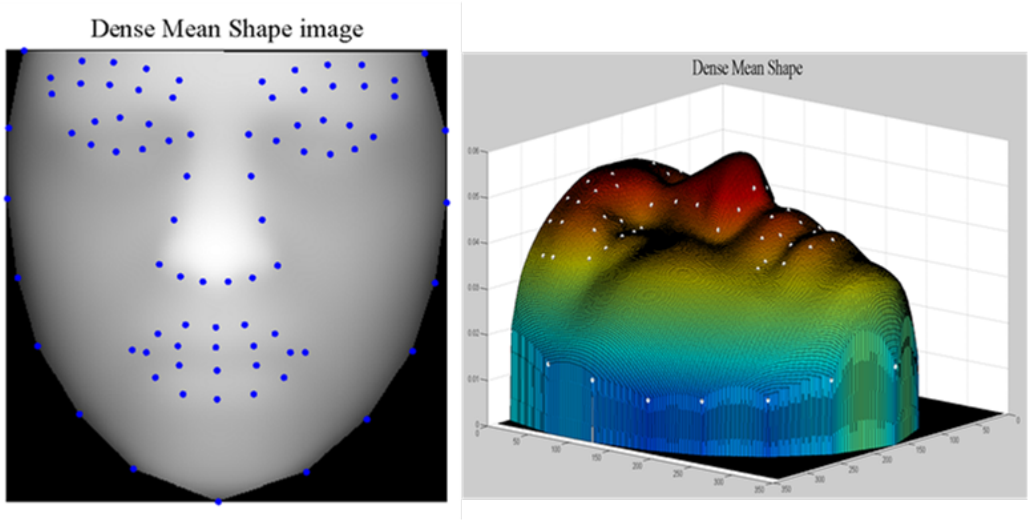


Fig6 Mean depth image

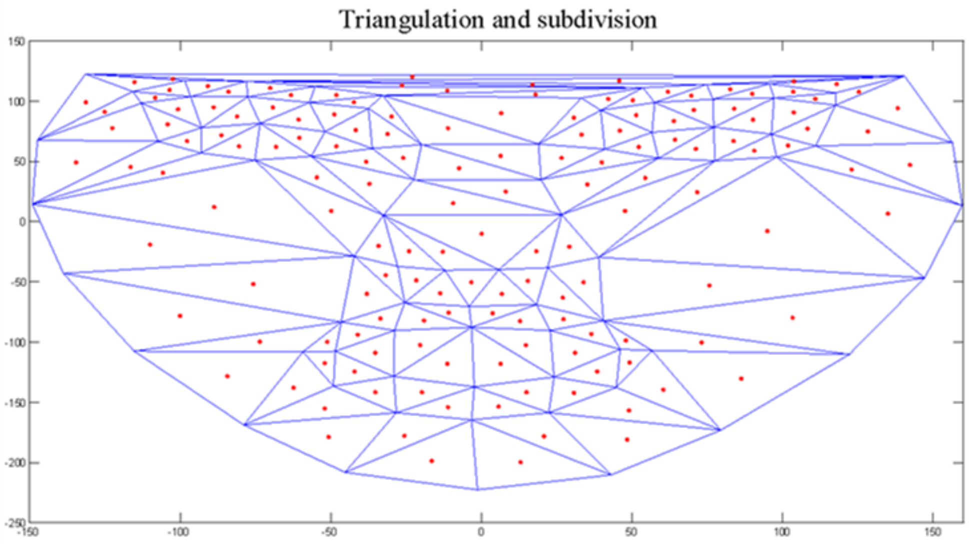


Fig7 Triangulation and Subdivision

[1]Generalized Procrustes Analysis

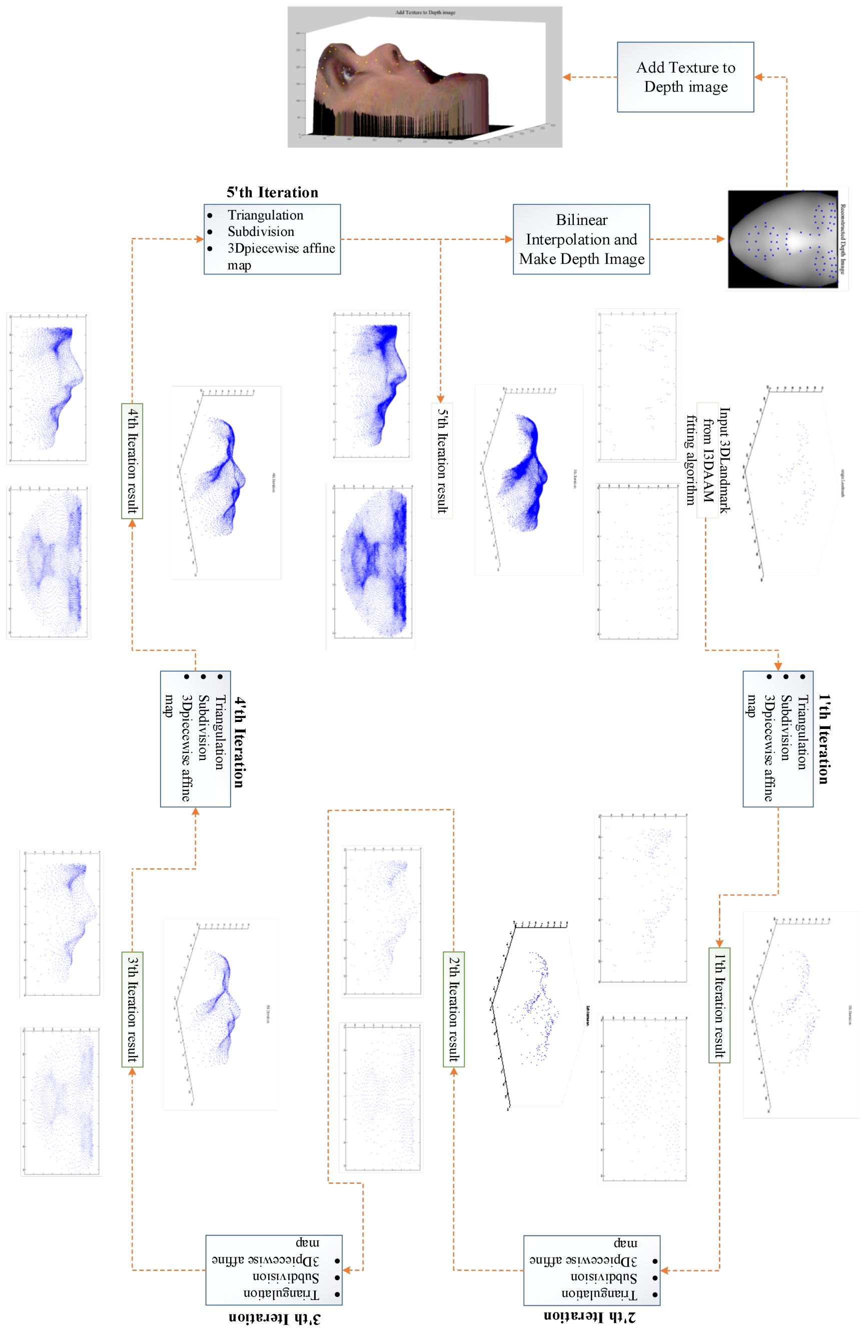


Fig8 Block diagram of subdivision and proposed 3D piecewiseaffine transform which convert sparse 3D landmarks to dense 3D point cloud

To achieve this goal we proposed an approach which applies subdivision and proposed 3D piecewise affine transform recursively to sparse 3D landmarks yield to convert sparse 3D landmark to dense 3D landmark and consequently depth image. In the first step, we triangulated 3D landmarks and selected the x and y coordinates of every triangle center as a new point with unknown z(fig7). Our goal is to find z and do it recursively to increase 3D points number. We assumed that every 3rd coordinate of every point(z) in each triangle could be calculated by a linear combination of its x, y coordinates and the 3rd coordinate of the corresponding point in the mean depth image. So we have three unknown which can be calculated with three equation generated by three vertices of the triangle with known z. Block diagram of the proposed approach is given in fig8. Fig9 illustrates every iteration of the process applying to a surprised 3D landmark as an example.

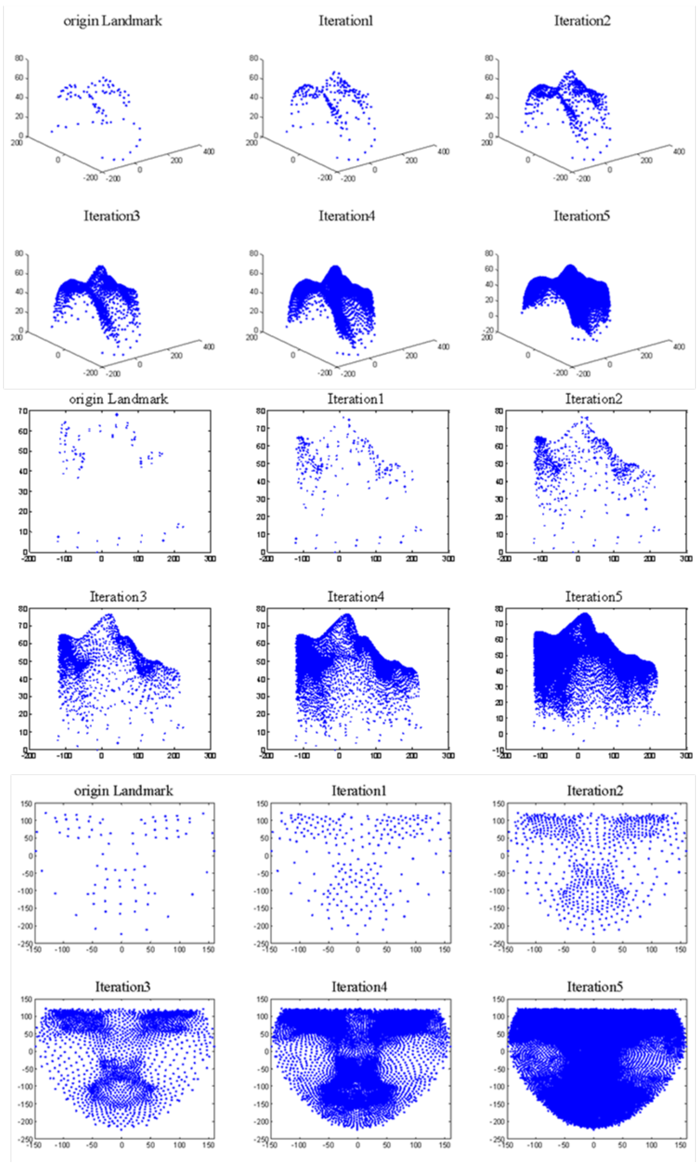


Fig9 Subdivision and proposed 3D piecewise affine iterations in multiple view

After then we create depth image from dense 3D point cloud generated in previous step(fig10) and finally mapped texture on it. Fig12 shows some 3D reconstruction with texture mapping. table1 shows that our proposed approach has less 3D reconstruction error comparison to [2]GEM that it is reasonable because GEM stretches a depth image according to the 2D landmark on the face image,

[2]Generic Elastic Model

but our proposed approach calculates every point's z in the new depth image.
In the following we extracted [3]3DLBP feature(fig11) from obtained depth image and showed that this feature yield better result in face recognition on partially blurred face image than [4]LBP and [5]LDN; because in these images texture is partially disrupted and these features can not discriminate faces from each other as so well as 3DLPB. Table2 , table3 and table4 are confusion matrix of 3DLBP, LBP and LDN respectively on seven face expression; so we can see that 3DLBP yields better performance in comparison to two other texture features on partially blurred face images.

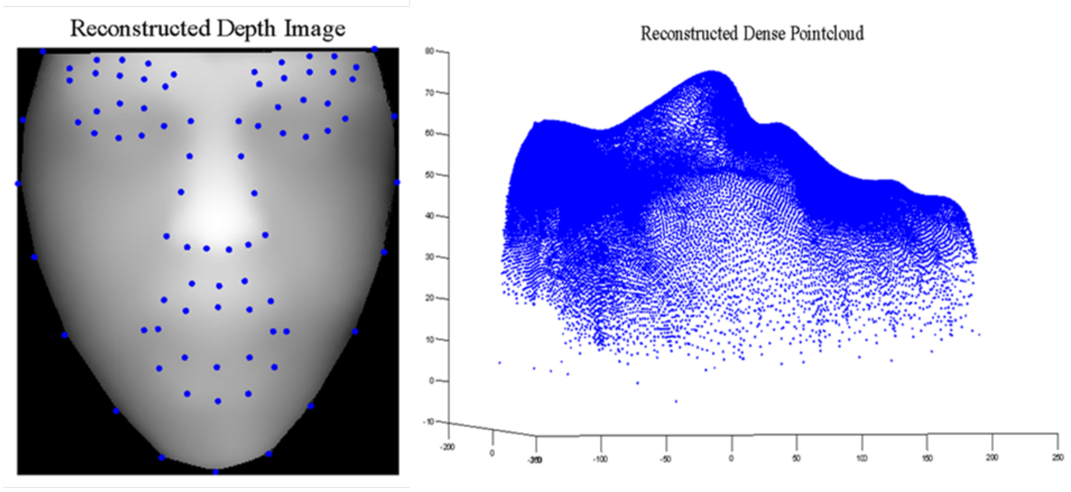


Fig10 Depth image creation from dense 3D point cloud

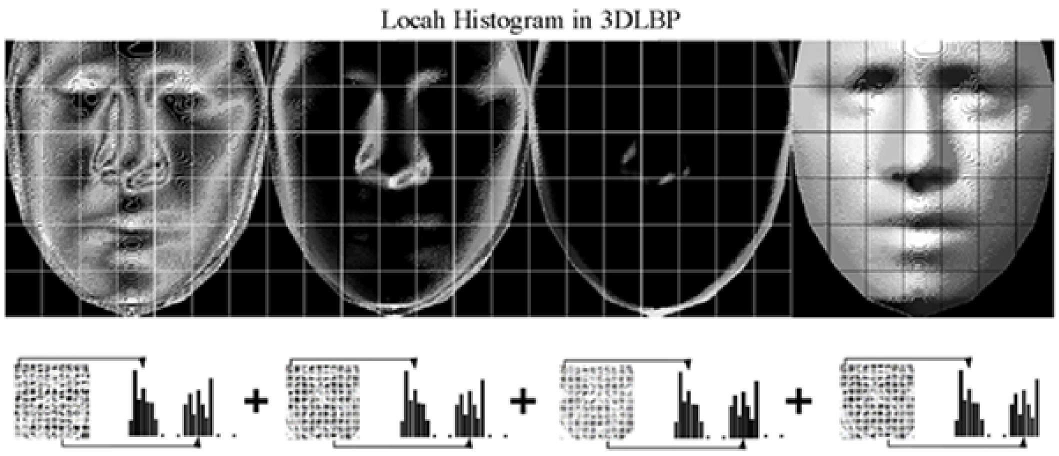


Fig11 3DLBP feature

[3]3D Local Binary Pattern
[4]Local Binary Pattern
[5]Linear Discriminant Analysis

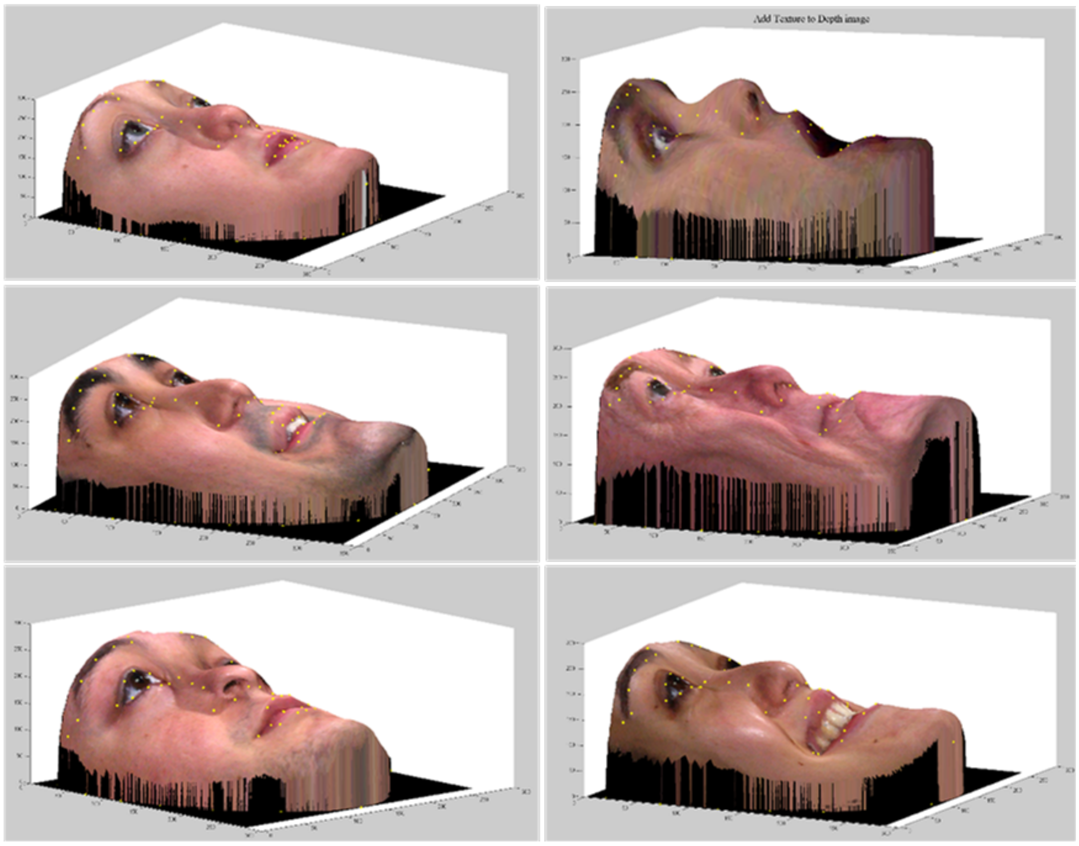


Fig12 Some 3D reconstruction with texture mapping

	GEM	**Our Proposed Approach**
Anger	10.03 ±7.82	7.06 ±5.93
Disgust	15.90 ±9.16	8.83 ±7.38
Fear	13.78 ±8.03	8.46 ±7.03
Happy	16.92 ±10.86	9.44 ±7.92
Neutral	10.15 ±4.62	6.20 ±4.82
Sadness	12.33 ±7.31	7.06 8.13±7.10
Surprise	18.64 ±12.19	10.79 ±9.26
mean	**13.96 ± 8.57**	**8.41 ± 7.062**

Table1 Mean and standard deviation of 3D reconstruction error for reconstructed depth image by GEM *&* Our proposed approach

	Ne(%)	**An(%)**	**Di(%)**	**Fe(%)**	**Ha(%)**	**Sa(%)**	**Su(%)**
Neutral	**81**	4	2	2	1	10	0
Angry	8	**58**	10	4	1	19	0
Disgust	1	9	**75**	7	3	3	2
Fear	4	6	9	**67**	6	6	2
Happiness	0	0	1	9	**89**	0	1
Sadness	11	12	9	9	1	**58**	0
Surprise	3	0	4	1	2	0	**90**
Average(%)	**74**						

Table2 Confusion matrix and face recognition rate in case of using **3DLBP** as feature extractor

	Ne(%)	**An(%)**	**Di(%)**	**Fe(%)**	**Ha(%)**	**Sa(%)**	**Su(%)**
Neutral	**74**	8	2	7	1	5	3
Angry	8	**51**	12	12	2	15	0
Disgust	2	10	**70**	9	2	4	3
Fear	4	12	8	**56**	10	6	4
Happiness	1	1	1	11	**86**	0	0
Sadness	12	23	1	6	2	**54**	2
Surprise	3	0	2	3	2	3	**87**
Average(%)	**68**						

Table3 Confusion matrix and face recognition rate in case of using **LBP** as feature extractor

	Ne(%)	**An(%)**	**Di(%)**	**Fe(%)**	**Ha(%)**	**Sa(%)**	**Su(%)**
Neutral	**28**	16	9	9	11	9	18
Angry	16	**25**	17	13	7	17	5
Disgust	11	12	**24**	13	13	10	17
Fear	10	8	14	**21**	25	12	10
Happiness	15	3	7	23	**40**	5	7
Sadness	17	20	13	15	4	**22**	9
Surprise	9	1	4	10	5	8	**63**
Average(%)	**32**						

Table4 Confusion matrix and face recognition rate in case of using **LDN** as feature extractor