3D Face Reconstruction

Mehmet Ozgur Turkoglu, *moturkoglu@gmail.com, s1814389, Electrical Engineering MSc.*

***Abstract*—In this work, I present a method for stereo-based 3D face reconstruction. The algorithm is based on depth estimation of the face by using image pair which are taken from different point of views. Afterwards, surface mesh of the face is generated by deploying this depth information. In order to improve the depth estimation, I take advantage of the face properties. Some experimental results provided show that the algorithm is promising.**

***Keywords*—3D face reconstruction, face depth estimation, stereo matching.**

I. Introduction

3D face reconstruction is a process of creating 3D model of the human face from 2D or 3D (2D image + depth image (laser data)) image or image pair. 3D face reconstruction is used in many applications such as plastic surgery simulations, face recognition, facial expression recognition, virtual reality simulations, 3D games and animations. Nowadays, 3D sensors which can be used for acquiring 3D faces are available; however these sensors are expensive and unpractical for many applications. Thus, accurately reconstructing the 3D face surface model from 2D images is still popular topic in computer vision.

Recently, there are several main approaches used for 3D face reconstruction from 2D image(s). 3D Morphable Models (3DMM) is a statistical model which usually consists of a PCA model of shape, color, camera and a lighting model and it requires a lot of training data. A set of shape and texture coefficients describes a face and these coefficients are determined from 2D image; for instance, [2] uses SIFT features to fit 3D Morphable Model. The other technique is warping to a reference shape which comprise to deform the pre-defined generic face model by using facial landmarks. [3] uses stereo images to detect 3D facial features and deform the face surface using Thin-Plate Spline method. If the images are from a video sequence, structure from motion (SfM) method is useful when key points are tracked across the face (e.g [4]).

The techniques I mentioned so far are more accessible since they do not require images which are taken simultaneously by two (or more) calibrated cameras but the main problem of those methods is that in general, they produce 3D face model which resembles generic model. So if we have images which are taken simultaneously by two calibrated cameras, multi-camera stereo (stereo matching) approach can produce more detailed, sufficiently subject-specific 3D face model. The method is based on finding the correspondences (matching pixels), that is the points in the images that are the projection of the same point in the scene and estimating distance between correspondences. However, this technique is more error-prone, can produce many outliers because even though, the idea is quite simple finding the correspondences in stereo pairs is problematic; therefore, stereo-matching has been the focus of much research in computer vision and it is still an open problem.

(a) (b) (c)

Fig. 1. Example of image pair from dataset

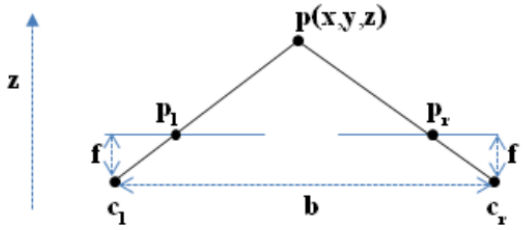


Fig. 2. Geometric model of calibrated stereo vision system. f:focal length, b:baseline, $p(x, y, z)$:3D point, p_l, p_r:pixel values of projected point on left and right image respectively. (Courtesy [1])

In this project, I tackle 3D face reconstruction based on stereo matching. In order to improve stereo matching and reduce outliers, I incorporate facial landmarks which can easily be detected.

The rest of the report is organized as follows. Section 2 provides a detailed description about my approach for 3D face reconstruction from 2D images, I mention about evaluation protocol and implementation details as well in this section. I will provide some results I obtained in Section 3. I will discuss finding I obtain and talk about some future works in Section 4. Finally, in Section 5 I will draw a conclusion.

II. Method

A. Methods

1) Background: When we have two fully aligned cameras (baseline is oriented horizontally (x-direction), the horizontal directions of both image planes are parallel to the baseline and the two optical axes are parallel (along z-direction) and they are orthogonal to the baseline.) (see Figure 2), we can estimate the depth of 3D point p according to similar triangles' principle.

$$\frac{z}{f} = \frac{x}{x_l} = \frac{x-b}{x_r}$$

$$d = x_l - x_r = \frac{fb}{z}$$

d is the disparity which is spatial shift of the point in the two images. Its value increases when the distance between the point p and the camera decreases. To estimate its value, it is necessary to find pixels in both images that correspond to the projection of the same real word point. This process is called stereo matching. When we know disparity of point p and camera parameters (b and f), we can find the depth using previous equation.

$$z = \frac{fb}{d}$$

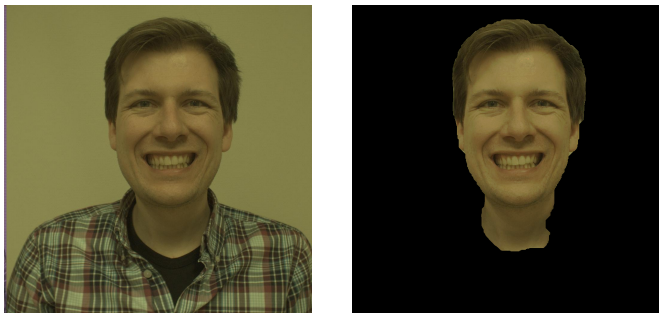
2) Face Segmentation: The input images provided for this project contain both face and background. So the first step of my approach is to segment face (face, hair and neck) of the subject. The images are not in-the-wild images, so the face segmentation is rather easy. I use color-based segmentation using k-means clustering. Input images consist of several major colors (e.g 3, 4). For instance, images of subject 1 has 3 major colors: color of skin, of hair and of background. I convert image to gray-level image and compute gray-level histogram of the image. This histogram is clustered into number of seeds (seed corresponds to centroid of the cluster) I used (The default number of seeds is 3, but it may change according to subject.). The number of seeds should be equal to number of major colors in the image. Then, the image is segmented accordingly. The pseudo-code for segmentation is given in Algorithm I. After color-based segmentation, I apply morphological operations to clear undesirible small parts. Figure 3 shows the result of the overall segmentation process.

- H = histogram(gray-scale image),
  $H = \{h^{(1)}, h^{(2)}, .., h^{(M)}\}$, M is the number of the bins in the histogram.
- Randomly initialize centroid of each cluster (seeds)
  $\mu_1\mu_2, ..\mu_K$, K is total number of clusters

while *not converge* **do**
  for *every i, set* **do**
    $c^{(i)} = argmin_j ||h^{(i)} - \mu_j||^2$
  end
  for *each j, set* **do**
    $\mu_j = \frac{\sum_{i=1}^{M} (c^{(i)}=j) h^{(i)}}{\sum_{i=1}^{M} (c^{(i)}=j)}$
  end
end

Algorithm 1: Face segmentation based on k-Means clustering

3) Histogram Equalization: "Constant Brightness Assumption" is a basic assumption that intensities of corresponding points in both images (stereo pairs) are equal. However, this assumption is generally false for stereo pairs due to lighting geometry and surface geometry etc. Consequently, if we want to compare two images, it is useful to give them first the same dynamic range and luminance. [5] proposed "midway image equalization" method which is more naive histogram equalization algorithm and shows high performance for stereo pairs. They generate a common histogram and this common histogram h stands midway between the previous histograms of u_1 and u_2, and as close as possible to each of them. This treatment must avoid to favor one histogram rather than the other.



(a) (b)

Fig. 3. Color-based face segmentation based on k-Means clustering.

This algorithm is based on equalization of cumulative histograms of image pairs. If u is the image defined on the discrete grid $\Omega = 0, ..., N-1x0, ..., M-1$, the cumulative histogram H_u of u is defined as

$$H_u(\lambda) = \frac{1}{|\Omega|} |\{(i,j) \in \Omega; u(i,j) \leq \lambda\}|$$

If u takes its values in the set $\{0, ..., L\}$, the discrete histogram h_u of u is the derivative of H_u as follow

$$h_u(l) = \frac{1}{|\Omega|} |\{(i,j) \in \Omega; u(i,j) = l\}|$$

Midway histogram of two histogram pairs is computed by taking the harmonic mean of them.

$$H_{midway} = \frac{2H_1H_2}{H_1 + H_2}$$

Then, images are transformed in such a way that their cumulative histogram become H_{midway} as following

$$u_1' = 0.5(u_1 + H_2^{-1} \circ H_1(u1))$$

$$u_2' = 0.5(u_2 + H_1^{-1} \circ H_2(u2))$$

Equalized image pairs by this method are shown in Figure 4.

4) Facial Feature Points Detection: Facial features (facial landmarks) are critical points on the face such as corners of the eyes, eyebrows, the mouth, the lip, the tip of the nose. These points provide a lot of information about the shape and the geometry of the human face. Since, recently detecting these landmarks can be done quite accurately, I decide to use them to improve disparity map (depth image).

I make 2 assumptions before using facial landmarks. My first assumption is there are enough facial landmarks and the

(a)

(b)

Fig. 4. Histogram Equalization using midway image equalization algorithm. The first row is input image pairs, and the second row is equalized image pairs.

(a) (b)

Fig. 5. Facial landmark detection. There are 68 facial landmarks which are connected by lines on the images.

set of landmarks include closest and farthest points of the face. My second assumption is that face surface is rather smooth.

I detect the facial landmarks by using Kazemi et. al. method [6] which is quite robust and fast model-based algorithm similar to Active Shape Models (ASMs). It uses 68 facial landmarks and these landmarks altogether are considered as a shape. It initialize the shape from mean shape (mean shape is obtained from training data) and update it by cascaded regressors which are trained with gradient tree boosting algorithm.

Detected facial feature points are shown in Figure 5.

5) Stereo Rectification: The cameras used for taking the stereo image pairs are not fully aligned so it makes harder finding the correspondences since epipolar lines are not know

(a) (b)

Fig. 6. Rectified stereo image pairs

(2D search problem). So in order to make the problem simpler, we transform the images (virtually rotate cameras by projective transform) such a way that as if two cameras are fully aligned. In this scenario, all the epipolar lines are horizontal on common image plane, so stereo matching (finding correspondences) becomes 1D search problem. Before applying projective transform to stereo image pairs, I find intrinsic and extrinsic camera parameters by using calibration images (checkerboard images). Matlab calibration tool also provides radial and tangential distortion parameters so I can correct nonlinear distortion as well. The rectified (and undistorted) image pairs are shown in Figure 6.

6) Sparse Disparity Map: After facial landmarks detection process, I have 2D coordinates of 68 face feature points in the right and left images. For each face feature point p_i , the Euclidean distance between its right and left coordinates is calculated to obtain its disparity $d(p_i)$ as follows

$$d(p_i) = \sqrt{(x_R - x_L)^2 + (y_R - y_L)^2} \approx |x_R - x_L|$$

Disparity of facial landmarks are shown in Figure 7. By considering my assumption (The set of landmarks include closest and farthest points of the face.), I determine the disparity range for the dense disparity map. So I reduce the search space; for instance, the range I found for the dataset images are in between 64 and 80 pixels which is quite small compared to image size (1024 by 1024). Reducing search space improves the performance of the dense disparity calculation highly.

$$diparity\ range = [min(d(p_i)), max(d(p_i))]$$

7) Dense Disparity Map: In this step, I calculate disparity of every points of the face. I use semi-global matching (SGM) algorithm to find disparities. Semi-global matching[7] combines concepts of global and local stereo methods for accurate, pixel-wise matching at low runtime. The energy which semi-global matching tries to minimize is following

$$\sum_{n,m\in\Omega} [G(n,m) + \sum_{i,j\in N(n,m)} T(D(n,m), D(i,j))]$$

where the first term ($G(n,m)$) is the cost for pixel is belong to certain disparity and the second term is the transition cost which provides smoothness in disparity map. The cost function is optimized similarly to dynamic programming. Since dynamic programming processes the image row by row, it ignores the fact that disparity is not only has context along the

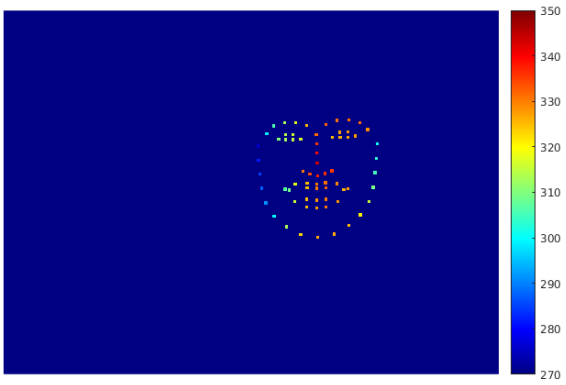


Fig. 7. Sparse disparity map (Disparities of facial landmarks). Pixels are depicted larger.

rows, but also in all other directions so it generates less smooth disparity map. The novel idea of SGM is the computation along several paths, symmetrically from all directions through the image. Each path carries the information about the cost for reaching a pixel with a certain disparity. For each pixel and each disparity, the costs are summed over the eight paths. Then, at each pixel, the disparity with the lowest cost is chosen. The disparity map calculated by SGM becomes more smooth compared to dynamic programming.

I also create disparity map by using simple correlation-based block matching method in order to enhance the final disparity map. The disparity maps are given in Figure 8.

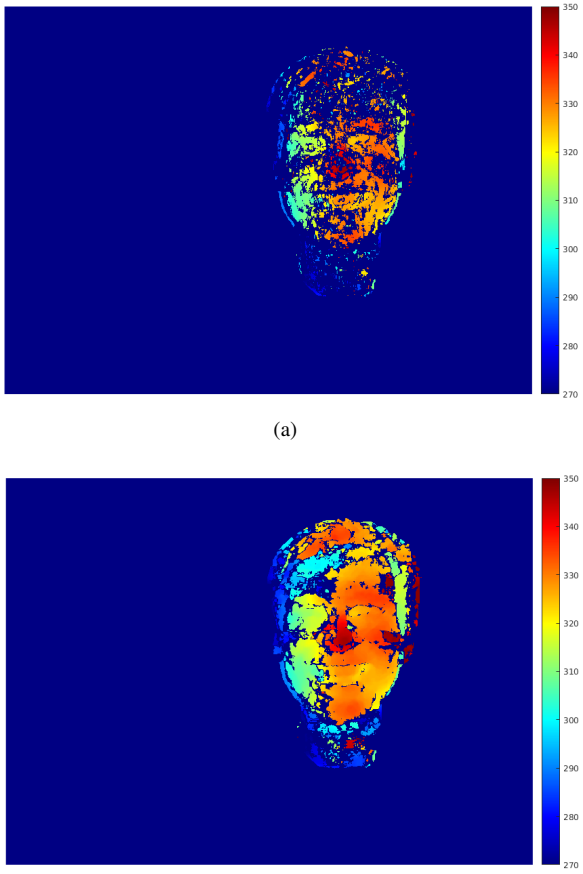
8) Outliers Detection: Because there are large less-textured areas in human face such as cheeks and forehead, the disparity map created by SGM has some erroneous regions (outliers). I detect these erroneous regions by incorporating sparse disparity map (disparity of facial landmarks).

To detect outliers, I first add 2 facial landmarks on each cheeks (because there is no facial landmarks on cheeks) to represent those regions and the disparity of these landmarks are determined from SGM disparity map by averaging. Then, I create dense disparity map from sparse disparity map by putting Gaussian on facial landmarks and the disparity of rest of the points are determined by these Gaussian distributions. I assume that the face surface is rather smooth so disparity map computed by SGM should be consistent with this disparity map. If the disparities of the pixel are not close to each other, I label that pixel as outliers and get rid of it from SGM disparity.

$$p_i \in outliers\ if\ |d_{SGM}(p_i) - d_{Gauss}(p_i)| > threshold$$

The disparity map computed from facial landmarks and the disparity map after getting rid of outliers are shown in Figure 9.

9) Post-processing: After outliers are discarded, I enhance the disparity map adding valid disparities form block-matching disparity map. The enhanced disparity map is shown in Figure 10. There are only a few pixels are changed but they are important especially in the region a lot of disparities are missing (e.g. right of the forehead and left of the cheek) when I interpolate (filling missing disparities) disparity image.



(a)

(b)

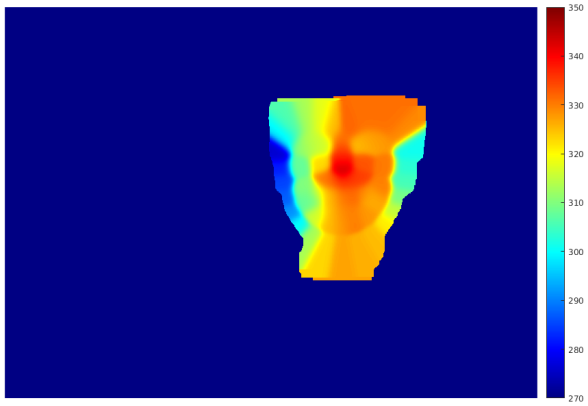
Fig. 8. Disparity maps of stereo image pairs. (a)Disparity map estimated using block matching. (b)Disparity map estimated using Semi-global matching algorithm.

I fill the missing pixels in the disparity image by using Gaussian mask. The interpolated image is shown in Figure 11.

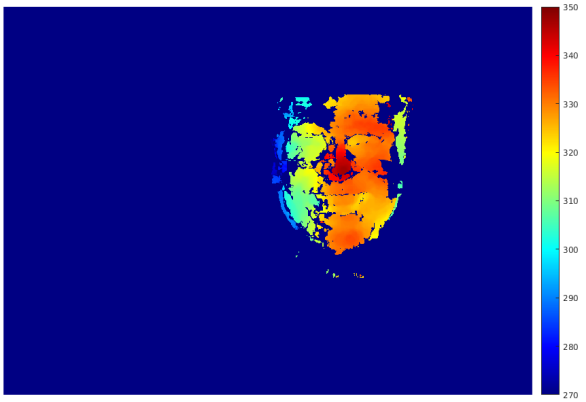
Afterwards, I apply median filter which is very crucial to suppress remaining outliers and smooth the face surface. Lastly, I smooth the face surface with Gaussian filter.

10)3D Surface Mesh: I already know camera parameters and I have depth image so I can create point cloud (3D position of the points). This point cloud is used for generating 3D face surface. 3D surface is created using Delaunay triangulation which is angle-optimal triangulation and maximize the minimum angle of all the angles of the triangles in the triangulation (Triangulation is a subdivision of a planar or higher-dimension geometric object into triangles). Reconstructed 3D surface of the face is given in Figure 12.

11)Performance evaluation: Since there is no ground truth for 3D face surface of input images, it is quite hard to evaluate the performance of my approach. However, 3 stereo image pairs are provided so I can evaluate my approach by comparing two 3D face surfaces which are reconstructed from different stereo pairs (This can be thought creating my own ground



(a)



(b)

Fig. 9. Outliers Detection. (a)Disparity map created from sparse disparity map (b)SGM disparity map after outliers are discarded

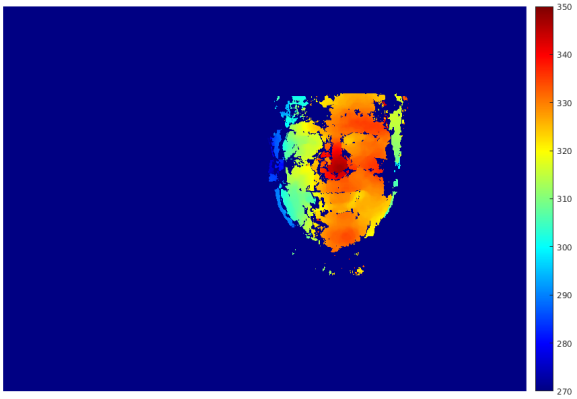


Fig. 10. Enhanced disparity map after adding valid disparities from block-matching disparity map on SGM disparity map.

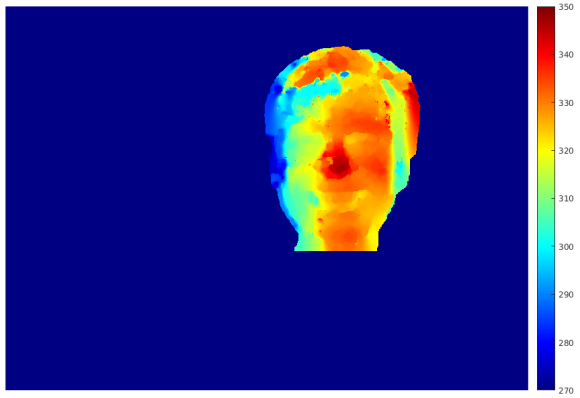
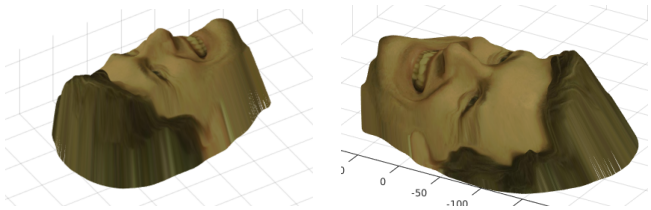


Fig. 11. Interpolated disparity image.



(a) (b)

Fig. 12. Reconstructed 3D face surface

truth). My evaluation protocol consists of following steps.
1-Reconstruct 3D surface from middle and right image pairs
2-Reconstruct 3D surface from left and middle image pairs
3-Compare 3D surfaces and if the same surface points are consistent, label that point correct.

In order to compare two different surfaces, I align two surface using Iterative Closest Point (ICP) algorithm. Afterwards, I compare each points and label them as correct if they are close enough to each other.

$$p = correct\ if\ d_{Euclidean}(p_1, p_2) < threshold$$

where p_1 and p_2 are the reconstructed 3D points of point p from first and second stereo pairs respectively. Determining threshold is problematic but we can choose it as 1% or 2% of human face size. The width of average human face is around 150 mm so threshold can be 1.5 mm and 3 mm.

Then, I calculate the overall accuracy of the system as following

$$accuracy = \frac{number\ of\ correct\ points}{total\ number\ of\ points}$$

B. Implementation Details

I use Matlab stereo calibration application in order to find intrinsic and extrinsic camera parameters. For facial landmark detection, I used Dlib C++ (Dlib is a modern C++ toolkit containing machine learning algorithms) library which includes Kazemi et. al. facial landmark detection algorithm. I used

	1	2	3	4	5
subject 1	0.3824	-	-	-	-
subject 2	0.1652	0.1725	0.2251	0.2019	0.2359
subject 4	0.3695	0.2965	0.3447	0.2878	0.3083

TABLE I. ACCURACY, THRESHOLD = 1.5 MM

	1	2	3	4	5
subject 1	0.6625	-	-	-	-
subject 2	0.4011	0.4025	0.4924	0.4401	0.5075
subject 4	0.6305	0.6090	0.6215	0.5665	0.5991

TABLE II. ACCURACY, THRESHOLD = 3.0 MM

Thierry Guillemot's midway image equalization implementation (C++) for histogram equalization. The rest of the pipeline is implemented in Matlab.

The Matlab implementation is available via following link: https://github.com/0zgur0/3D-face-reconstruction-/tree/master

III. RESULTS

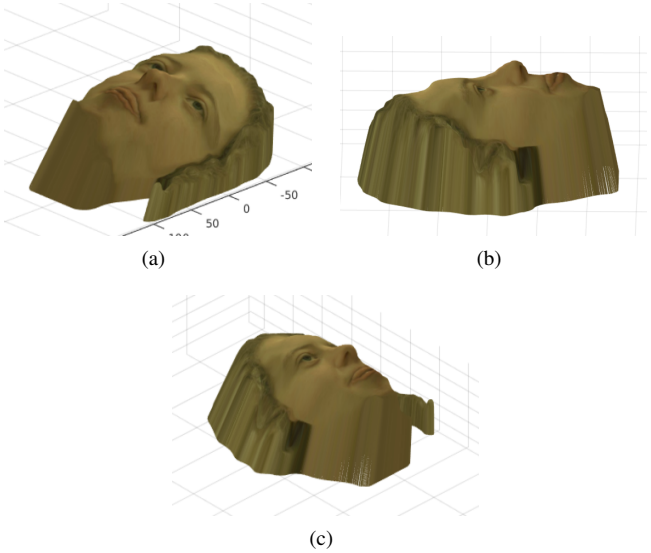
We have 11 stereo image pairs provided (subject 1:1, subject 2: 5, subject 4: 5). I made an experiment for them and calculate the accuracy for each pairs. The results are given in Table I and II.

IV. DISCUSSION

The most challenging part of this project is estimating disparity for texture-less regions of the face such as forehead (see Figure 14), because stereo matching algorithms are more error prone when there is less texture information. Even though, cheeks have less texture, they were reconstructed better because outliers were detected by using facial landmarks. So if we use more facial landmarks (landmarks on forehead as well) we can reconstruct face especially forehead more successfully.

Finding disparity of hair is erroneous. Even though, 3D face reconstruction of subject 2 (see Figure 13) is better than of subject 4 (see Figure 14), accuracy for subject 4 is higher due to hair of subject 2.

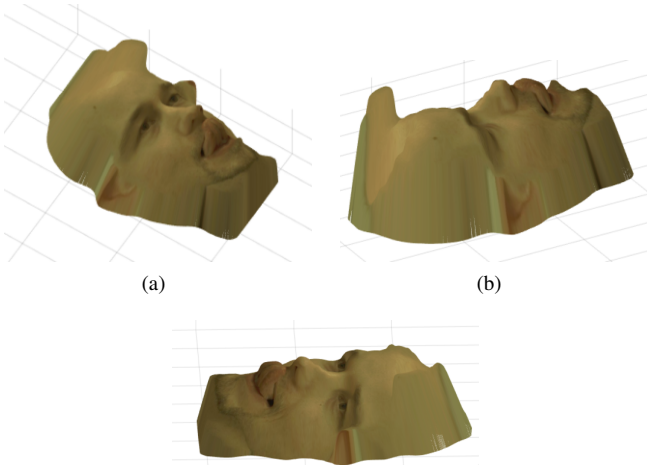
I try to estimate dense disparity map from sparse disparity map (see Figure 9(a)). When I do this I put same Gaussian distribution (same standard deviation) on each facial landmark, better estimation of that map might be done by finding the optimal covariance matrix for each Gaussian.



(a) (b)

(c)

Fig. 13. Example of reconstructed 3D face: Subject 2



(a) (b)

(c)

Fig. 14. Example of reconstructed 3D face: Subject 4

V. CONCLUSION

In this project I worked on the 3D face reconstruction from stereo image pairs. To reconstruct the depth form stereo pairs, we have to find correspondences (stereo matching) which is not trivial. Because some parts of the human face have less texture information, it makes the stereo matching problem more sophisticated. I incorporate facial landmarks to improve the performance of stereo matching. The results I obtain are almost reasonable but the method can be improved easily some quick modification such as increasing number of facial landmarks.

REFERENCES

[1] Aissaoui, Amel, et al. *"Fast Stereo Matching Method based on Optimized Correlation Algorithm for Face Depth Estimation."* VISAPP (2). 2012.

[2] Huber, Patrik, et al. *"Fitting 3D morphable face models using local features."* Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015.

[3] Choi, Jongmoo, et al. *"3D face reconstruction using a single or multiple views."* Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010.

[4] Chowdhury, Amit K. Roy, and Rama Chellappa. *"Face reconstruction from monocular video using uncertainty analysis and a generic model."* Computer Vision and Image Understanding 91.1 (2003): 188-213.

[5] Delon, Julie. *"Midway image equalization."* Journal of Mathematical Imaging and Vision 21.2 (2004): 119-134.

[6] Kazemi, Vahid, and Josephine Sullivan. *"One millisecond face alignment with an ensemble of regression trees."* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[7] Hirschmuller, Heiko. *"Accurate and efficient stereo processing by semi-global matching and mutual information."* Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 2. IEEE, 2005.