Three Dimensional Quantitative Evaluation of Asymmetry in Facial Expressions

Pujitha Gunaratne, Yukio Sato
Department of Electrical and Computer Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan
pujitha@hilbert.elcom.nitech.ac.jp, sato@elcom.nitech.ac.jp

Abstract

We present a simple but effective approach to estimate the asymmetry in facial expressions using range data. A face measuring system called "Cubicfacer" that equips two laser scanners and a color CCD camera measures the human faces and produces range and color texture images simultaneously. Five pre-determined facial actions are measured on each subject and 42 symmetrically distributed mapping points are then extracted from the color images of each action. Pre-selected nodes of a generic face mesh are mapped to these points using a least squares approximation method. A local approximation for facial parts is then applied iteratively and the measured range data is transformed to produce a complete wrap of the mesh. This procedure is applied for each measured facial action and the deformation obtained is analyzed with sub-meshes that correspond to forehead, eyes, mouth, cheeks and chin areas. Variance and regression analysis are performed on patch pairs on left and right halves of the meshes to estimate the asymmetry. We set a threshold value to extract the prominent patches with greater deformation variances to produce an estimation of expression asymmetry. Error estimation is further applied to analyze the mesh fitting anomalies.

1. Introduction

The interests of modeling human faces to be used in diverse application areas span over three decades. The primary goals of the early work were to animate and produce CG characters from the data obtained by 2D or 3D image acquisition techniques. Development of measuring systems, which produce accurate and dense 3D data, take the modeling and interpretation tasks to new heights due to the availability of geometric information of the face. The work on facial expression analysis directly applied in the areas of face recognition, morphing, simulation models, animations and medical applications.

Although the human face seems to be a symmetric entity, it would be hard to find humans with identical facial symmetry. To our evaluation model, inputs are not only the faces of *apparent-symmetric*, but also individuals with expression disorders. In this work our attention is primarily focused on analyzing different facial actions and producing a quantitative description to evaluate the asymmetry of each individual. The outcome is to be used as a yardstick in facial expression disorder analysis in identification, recognition and treatment application areas.

Various approaches of generic face mesh deformation are proposed in the past that applied in parameterized and control point models [1,2,3], spline based model [4] etc. In most cases, mesh adaptation require segmentation of underlying 3D surface or setting up control points on feature boundaries, generating overheads in processing. An integration of optical flow techniques with mesh adaptation is proposed in [5], where deformation parameters are needed to setup in the adaptation. The face modeling system proposed in [6] automatically generates topological face mesh by applying adaptive mesh techniques employed in [7]. Although it generates a symmetric model, number of patches in each measured action tends to vary, making it hard to use in consecutive expression analysis. In this work we employ a generic face mesh with known topology, and use an *N*-degree polynomial approximation with a least squares method to generate a wrap of the mesh to the measured range data.

2. The Approach

In our approach we measure each individual with five different facial actions using the "*Cubicfacer*", which measures a human face with two laser range scanners mounted on lateral sides of the face and a CCD camera in between them, facing the measuring profile (fig.1).

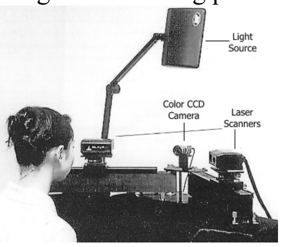
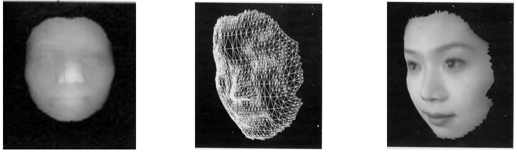


Figure 1. Face measuring system - Cubicfacer.

A complete 3D face model is produced in less than a second [8]. An adaptive mesh generation is then applied to range data after smoothing by median filtering, followed by texture mapping, to produce a realistic face model as depicted in fig.2. Since we use a single camera with same orientation in the measurement, both range and color images maintain a 1:1 correspondence between them.



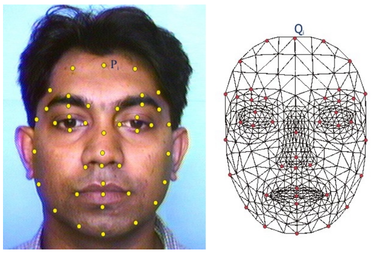
(a) (b) (c)

Figure 2. 3D Face model.
(a) Median filtered range image (b) Adaptive mesh result (c) Texture mapped result

In the phase of texture mapping, adaptive mesh generation is applied to the range data. Since the mesh generated by adaptive mesh does not possess consistent triangle density on both sides of the face, it is not suitable for interpretations based on symmetry features of the face. Therefore, we adopt an arbitrary generic mesh (fig.3-b), that is symmetric along the median plane, which is the vertical plane passes through the center of the nose, cutting the face into identical left and right halves. This generic mesh is in the 2D from, lies on the XY plane. Thus, we adopt a method of wrapping the mesh on to the measured 3D range data with the use of the corresponding color texture image.

2.1 Mesh adaptation

The mesh adaptation can be time consuming, tedious process if it involves segmentation of range data to extract features. Instead, here we apply a simple method of extracting features by using the corresponding color image, since it possesses the property of 1:1 correspondence with the range image. We apply sobel filter to the color image and extract prominent boundaries in eyes, nose and mouth regions by binarizing the sobel image. The nose tip location is obtained simply by searching the maxima in the range image. Since the facial outline obtained by the texture image does not adapt to any predefined shape, we use manual mouse clicks on the contours to generate outline points. We extract 42 distinct points from the texture image (fig.3-a), whose locations are known with respect to the generic mesh, for the adaptation of the face mesh.



(a) (b)

Figure 3. Generic mesh adaptation.
(a) Texture points - $\mathbf{P_i}$ (b) Mapping mesh nodes - $\mathbf{Q_i}$.

Points extracted from the color image are then polled to their mesh counterparts by a least squares approximation of a *N*-degree polynomial function.

2.2 Least squares approximation

We use an N^{th} order polynomial function for the least squares approximation to fit the generic mesh to the measured data. This process consists of two steps. First we move mesh vertices to the extracted points of the color image, and then poll the *Z* values from the corresponding range image.

Let us consider the parametric function given by,

$$Z = f(x, y).$$

Where, $f(x, y)$ represents by a polynomial of N^{th} degree, given by,

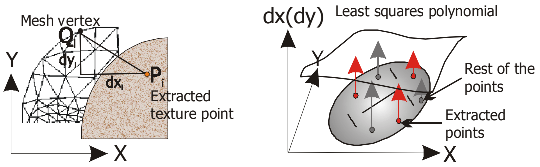
$$f(x, y) = a_{00} + \sum_{j=1}^{N} \sum_{i=0}^{j} a_{j-i,i} x^i y^{j-i} \text{ ---------------------(1)}$$

When *N*=2, it takes the form,

$$f(x, y) = a_{00} + \sum_{i=0}^{1} a_{1-i,i} x^i y^{1-i} + \sum_{i=0}^{2} a_{2-i,i} x^i y^{2-i} \text{ ------(2)}$$

Now consider match points P_i and Q_i, where *i*=1….*n*, represent *n* points on the color image and the mesh respectively (Fig.3). P_i's are extracted from the color image and Q_i's are known with respect to the generic mesh. Let (x_{P_i}, y_{P_i}) and (x_{Q_i}, y_{Q_i})represent 2D coordinates of P_i and Q_i respectively. We can thus calculate the displacement vectors, $\mathbf{dx_i} = (x_{P_i} - x_{Q_i})\vec{\mathbf{m}}$ and $\mathbf{dy_i} = (y_{P_i} - y_{Q_i})\vec{\mathbf{n}}$, where $\vec{\mathbf{m}}$ and $\vec{\mathbf{n}}$ are unit vectors along *x* and *y* direction respectively, for all matching points *i*=1,…,*n*.

Since we extract 42 points for initial matching, *n* is set to 42. To calculate displacement vectors for rest of the mesh points, we approximate the parametric function given in eq.(1) using the least squares method, polling $\mathbf{dx_i}$ and $\mathbf{dy_i}$ in *Z* axis as depicted in fig.4.



(a) (b)

Figure 4. Least squares polynomial approximation.
(a) Displacement vector (b) Polynomial approximation

In the first instance we apply a second degree polynomial function as in eq.(2). Thus, coefficients a_{00}, a_{10}, a_{01}, a_{20}, a_{11}, a_{02} can be calculated using initial displacement vectors.

The uncertainty of fitting can be calculated by the error term of least squares approximation as,

$$\mathrm{E} = \sum_{i=1}^{n} \left[f(x,y) - Z_i \right]^2 .$$

For the best fit, E must be minimum. Thus, for a second order polynomial,

$$\frac{\partial \mathrm{E}}{\partial a_{00}} = 0 , \frac{\partial \mathrm{E}}{\partial a_{01}} = 0 , \cdots\cdots\cdots\cdots , \frac{\partial \mathrm{E}}{\partial a_{20}} = 0 \cdot$$

Hence,

$$\frac{\partial \mathrm{E}}{\partial a_{00}} = \sum_{i=1}^{n} \left[f(x_i, y_i) - Z_i \right] \times 1^{00} = 0$$

$$\frac{\partial \mathrm{E}}{\partial a_{01}} = \sum_{i=1}^{n} \left[f(x_i, y_i) - Z_i \right] \times x_i^{01} = 0$$

$$\vdots$$

$$\frac{\partial \mathrm{E}}{\partial a_{kl}} = \sum_{i=1}^{n} \left[f(x_i, y_i) - Z_i \right] \times y_i^{k} x_i^{l} = 0$$

Where, $y_i^k x_i^l$ denote the derivative of $y_i x_i$ with respect to the a_{kl} . Simplifying above equations, and applying in eq.(2) we can calculate coefficients of the polynomial, $\mathbf{A}^{\mathrm{T}} = \left[a_{00}\ a_{10}\ a_{01}\ a_{20}\ a_{11}\ a_{02} \right]$, by solving the matrix equation,

$$\begin{bmatrix} \sum 1 & \sum x_i & \sum y_i & \sum x_i^2 & \sum y_i x_i & \sum y_i^2 \\ \sum x_i & \sum x_i^2 & \sum y_i x_i & \sum x_i^3 & \sum y_i x_i^2 & \sum y_i^2 x_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum y_i^2 & \sum y_i^2 x_i & \sum y_i^3 & \sum y_i^2 x_i^2 & \sum y_i^3 x_i & \sum y_i^4 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ \vdots \\ a_{02} \end{bmatrix} = \begin{bmatrix} \sum Z_i \\ \sum Z_i x_i \\ \vdots \\ \sum Z_i y_i^2 \end{bmatrix}$$

Hence, displacement vectors $\mathbf{dx_i}$, $\mathbf{dy_i}$, of all mesh points, where *i*=1…..,.*M*, can be calculated by interpolating with coefficients $\mathbf{A}^{\mathrm{T}}$. *M* is the total number of points in the face mesh.

We repeat this procedure again by increasing the order of the polynomial to a forth order, and setting a threshold value for the error factor. The idea is to move the mesh points further closer to the expected locations by iterative approximation.

We then separate feature points on different regions of the face, namely eye, nose and mouth regions, where a high concentration of facial features is observed.

Suppose, *P'i* and *Q'i* are the match points on a particular region, representing extracted color image points and the mesh points, respectively. Let their current displacements, after the initial mapping, be $\mathbf{dx_j}, \mathbf{dy_j}$, where *j*=1,..,*m*, with *m* extracted points. Then we do the same calculations for these local values to generate a localized mapping further for the selected feature regions, using region dependant threshold values. This local matching is done to ensure a better mapping for the prominent feature areas of the face. Finally, *Z* values are mapped from the corresponding range values, since both color and range images have 1:1 correspondence, producing a complete wrap of the face mesh to the measured 3D data.

The steps that involved in mesh fitting process can be summarized as follows.

Step (1): extract *n* mapping points from the captured color image. (*n* = 42 in our case.)

Step (2): calculate the displacement vectors $\mathbf{dx_i}$, $\mathbf{dy_i}$, *i*=1,..,*n* of those extracted points and corresponding mesh counterparts.

Step (3): apply polynomial function of second degree for the least squares estimator, calculate coefficients $\mathbf{A}^{\mathrm{T}}$.

Step (4): interpolate the displacements of other mesh points using the coefficients $\mathbf{A}^{\mathrm{T}}$.

Step (5): compare fitting error of all mesh points with a threshold value.

Step (6): if the error exceeds the threshold, calculate new displacement vectors $\mathbf{dx_i}, \mathbf{dy_i}$, *i*=1..,*n*, with respect to the generated mesh.

Step (7): increase the order of the polynomial function for the least squares estimator, and calculate new coefficients $\mathbf{A}^{\mathrm{T}}$.

Step (8): repeat steps (4) to (6). If error still exceeds threshold, continue to step (7). Else,

Step (9): separate extracted points according to eyes, nose and mouth regions. Calculate their displacement vectors $\mathbf{dx_j}, \mathbf{dy_j}$, *j*=1…*m,* where *m* is the number of extracted points in each region.

Step (10): repeat steps (3) to (7) with calculated local displacement vectors in step (9).

Step (11): When the threshold is satisfied, map the corresponding range values from the measured range image to produce a complete 3D wrap of the mesh.

Once the 3D mesh is generated, we apply asymmetry measurements against the measured facial actions to estimate the difference of deformation on both sides of the face.

3. Estimation of Facial deformation

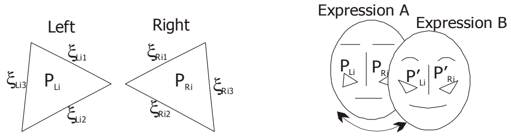
Facial deformation is estimated by calculating variances of patches on both sides.

3.1. Patch variance – single action.

Patch variances are estimated with respect to the sub-meshes representing different regions of the face. Forehead, eye, nose and mouth meshes are defined in the generic face mesh beforehand, and used to estimate the patch variances. Consider two matching patch pairs in a given sub-mesh marked as $\mathrm{P_{Li}}$ and $\mathrm{P_{Ri}}$, representing left and right side patches respectively (fig.5-a). Their corresponding edge lengths are denoted as ξ_{Li} and ξ_{Ri}, where *i* = 1,2,3, respectively. If the variance of the *i*th patch is ${\sigma_i}^2$ and total path variance of a given sub-mesh with N patches is σ^2, we can define,

$$\sigma^2 = \sum_{i=1}^{N} {\sigma_i}^2 = \sum_{i=1}^{N}\sum_{j=1}^{3} \left\| \xi_{Li_j} - \xi_{Ri_j} \right\|^2 .$$

Thus, we can measure the variance in terms of sub-mesh patches.



(a) **(b)**

Figure 5. Variance computation.
(a) single action (b) multiple actions

3.2 Patch variance – multiple actions.

In the applications of asymmetric facial expression analysis, it is often required to measure the variances of different facial actions and compare them (Fig.5-b). In a similar calculation, as done in the previous case, suppose patch P_{Li} of expression A occupies the patch P'_{Li} in expression B. Let the patch variances of left and right sides denote ${\sigma_L}^2$ and ${\sigma_R}^2$ respectively. Thus,

$${\sigma_L}^2 = \sum_{i=1}^{N_L}\sum_{j=1}^{3} \left\| \xi_{Li_j} - \xi_{L'i_j} \right\|^2 \text{ and } {\sigma_R}^2 = \sum_{i=1}^{N_R}\sum_{j=1}^{3} \left\| \xi_{Ri_j} - \xi_{R'i_j} \right\|^2$$

where ξ_L, $\xi_{L'}$, ξ_R, $\xi_{R'}$ represent the lengths of the same patch in right and left sides in different expressions.

N_L and N_R represent the number of patches in left and right sides of the same sub-mesh. Then the comparison is performed for both sides of the face to detect the asymmetry.

4. Quantitative analysis

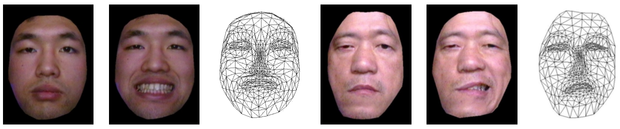
In this work, we measure five different facial actions of different human subjects with no apparent expression disorders, as well as subjects with some expression disorders. The actions are chosen to cover movements of most parts of the face. These are categorized in the Table 1. We analyze the range data of human subjects measured by the cubicfacer range finder system, using above described variance estimations.

Action code	Facial deformation
A01	Lines on the forehead.
A02	Eyes closed
A03	Sniff
A04	Grin
A05	Lip purse

Table.1: Facial actions analyzed

4.1 Variance analysis – single action

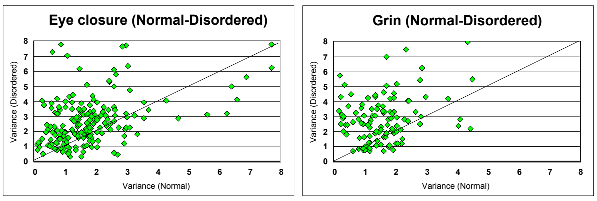
Variance of left and right side patches of sub-meshes corresponds to eye, mouth, nose and forehead regions are calculated to determine the asymmetry in measured actions. Measurement result of a normal subject and a one with expression disorder is depicted in fig.6.



(a) (b) (c) (d) (e) (f)

Figure 6. Measured actions: normal and disordered subjects.
(a) Relaxed-normal (b) A04 – normal (c) A04 mesh - normal (d) Relaxed – disordered (e) A04 – disordered (f) A04 mesh – disordered.

The patch variance is calculated in two ways. First, for each facial action, similar patches on left and right side are evaluated and the variance (σ^2) is calculated as discussed in section 3.1. Variances of forehead, eye, nose and mouth regions are calculated for normal and disordered subjects and disordered variations are plotted against normal variation sets. The results of eye closure and grin expressions are depicted in fig 7.



(a) (b)

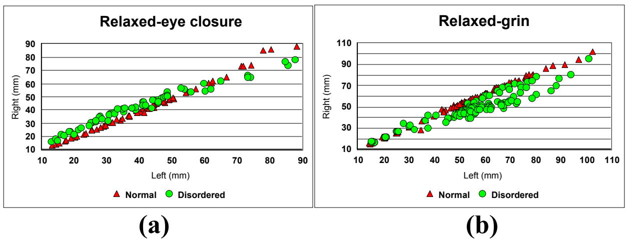
Figure 7. Variance comparison- disordered against normal. (a) Eye closure (b) Grin.

It can be seen that the bulk of the data has shifted towards the disordered axis indicating high patch variance with respect to the similar normal counterpart.

4.2 Patch variance – multiple actions

In this case, the variance calculation is done for an identical patch in two different actions. Let ${\sigma_L}^2$ and ${\sigma_R}^2$ represent the left and right patch variations respectively. They are compared to detect asymmetry, as described in section 3.2. We set a threshold value to prune the smaller variances and extract the higher ones, and graphically display the significant deviations in each action. Regression of left and right side patch variances are also calculated. The standard error factor gives the indication of deviation of data from its ideal expectations.

In fig.8 we demonstrate the variances of eye closed and grin actions relative to the relaxed situation for normal and disordered subjects.



	Standard error	
	A02	*A04*
Normal	1.342822	1.436317
Disordered	2.850067	3.441587

(c)

Figure 8. Eye closure and grin comparisons with respect to relaxed condition. (a) Relaxed-eye closure (b) Relaxed-grin.(c) Standard error in both cases.

5. Error compensation for mesh fitting.

As we demonstrated up to now, our estimations have a significant impact on the mesh fitting procedure described earlier in section 1.1. Therefore here we implement an evaluation method, to weigh the impact of fitting error. Here we measure a human subject with 50 markers attached on pre-determined locations on each face. The selection of markers is carefully designed so that they do not coincide with feature points and mostly belongs to non-feature areas of the face. This enables us to estimate the error components of freely moving parts especially on the cheeks and mouth areas. Similar expressions described in the table 1 of section 4 are measured and the same fitting process is applied. Then we track down the markers in each expressed face against the relaxed face to determine whether the markers occupy the same patches in both cases as expected in an ideal situation. Marker positions are extracted from the color image and their locations are recorded.

5.1 Estimation of patch movement

Although the coordinate systems change between two captured actions, local measurements with respect to the patches are invariant of each coordinate system. Therefore, marker locations are determined using non-zero, localized scalar parameters with respect to the patches they belong.

We first project the generated 3D mesh wrap in to X, Y plane since we extract markers from 2D color images. Thus, at each estimation, direct calculations can be applied with out transforming vector coordinates to the three dimensional space. Suppose marker M_i belongs to the patch P_i in the generated 3D mesh in the expression E_i, after mapping, as depicted in fig.9.

Suppose position vectors of vertices of patch P_i are given by $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$ respectively. Let vectors $\mathbf{a} = \mathbf{v_1} - \mathbf{v_3}$ and $\mathbf{b} = \mathbf{v_2} - \mathbf{v_3}$. Let λ and μ are scalar parameters and the position vector of marker M_i is represented by $\mathbf{m}$, where,

$$\mathbf{m} = \mathbf{v_3} + \lambda\mathbf{a} + \mu\mathbf{b} - - - (I),$$

for all $\lambda > 0$, $\mu > 0$ and $\lambda + \mu < 1$, so that marker M_i always occupies the patch P_i.

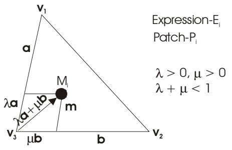


Figure 9. Position of marker M_i in patch P_i.

We also let a_x, a_y and b_x, b_y represent the x and y coordinates of vectors $\mathbf{a}$ and $\mathbf{b}$ respectively. If m_x, m_y and v_{3x}, v_{3y} are the respective x and y coordinates of vectors $\mathbf{m}$ and $\mathbf{v_3}$, from (*I*), we can derive,

$$\lambda = \frac{b_y(m_x - v_{3x}) - b_x(m_y - v_{3y})}{(a_x b_y - a_y b_x)} \text{ and } \mu = \frac{a_x(m_y - v_{3y}) - a_y(m_x - v_{3x})}{(a_x b_y - a_y b_x)}.$$

Hence we can calculate the location of marker M_i with respect to the patches on each sub-mesh for every measured facial action. In an ideal fitting situation, marker M_i would always occupy the same path in the same sub-mesh with identical λ and μ values. But it hardly happens in practice, so that we calculate the difference to determine the amount of shift each marker gets in different facial actions.

Now suppose, marker M_i occupies the patch P_i in expression E_i and it shifts to patch P_j in expression E_j, as depicted in fig.10.

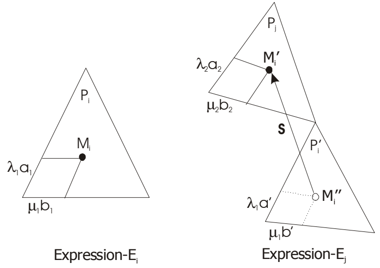
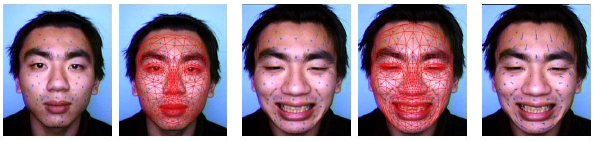


Figure 10. Marker shift with respect to patches in expressions.

Let marker location vectors in expression E_i be $\lambda_1\mathbf{a_1}$ and $\mu_1\mathbf{b_1}$. Similarly, that of in expression E_j be $\lambda_2\mathbf{a}_2$ and $\mu_2\mathbf{b}_2$. Lets consider if patch has not been shifted, as in the ideal case. Then we expect M_i to be in patch P'_i, which is P_i in expression E_j as shown in fig.10 with marker position M_i''. Thus, we can calculate the shift vector $\mathbf{S}$ for each marker and the movement of patch it belongs to, in different expressions. Fig.11 shows the human subject measured with markers attached.



(a) (b) (c) (d) (e)

Figure 11. Patch movement with markers (a) Relaxed (b) Relaxed mesh wrapped (c) Grin (d) Grin mesh wrapped (e) Shift vector.

The magnitude of shift vector $\mathbf{S}$ in fig.11 (e) indicates the amount of movement in patches in different locations of the face.

5.2 Error processing

Shift errors of patches occur due to the failure of patches to occupy the identical positions expected in different expressions. This error cannot be totally eliminated as in the ideal case due to many reasons in practice. One reason is that we cannot expect the extracted feature points in different actions to lie on the identical positions on the face all the time. Therefore, here we apply an iterative fitting solution to remedy the shift error, by applying fitting algorithm repeatedly with shift vectors.

Since shift vectors indicate the magnitude of shift of the patches concerned, we can treat them as displacement vectors that we described previously in the section 2.2 under the least squares approximation of the mesh fitting algorithm.

Let $\mathbf{m_i'}$ and $\mathbf{m_i''}$ represent the position vectors of M_i' and M_i'' (fig.12) and $x_{\mathbf{m_i}'}, y_{\mathbf{m_i}'}$ and $x_{\mathbf{m_i}''}, y_{\mathbf{m_i}''}$ represent their X, Y coordinates respectively. Since vector $\mathbf{S}$ can be represented as; $\mathbf{S} = \mathbf{m_i'} - \mathbf{m_i''}$, displacement vectors $\mathbf{dx_i}$ and $\mathbf{dy_i}$ can be defined as, $\mathbf{dx_i} = (x_{m_i'} - x_{m_i''})\vec{\mathbf{m}}$ and $\mathbf{dy_i} = (y_{m_i'} - y_{m_i''})\vec{\mathbf{n}}$, where $\vec{\mathbf{m}}$ and $\vec{\mathbf{n}}$ represent the unit vectors along x and y directions respectively. Thus, repetitive least squares algorithm is applied with initial polynomial of order 2 and increasing gradually as described previously in section 2.2 for smooth mesh fitting. Once the coefficients $\mathbf{A}^{\mathbf{T}}$ are calculated (described in section 2.2), each sub-mesh is moved accordingly to compensate the shift in patches. We repeatedly applied the fitting algorithm for 10 times and the resulting magnitude of shift vector between relaxed and grin actions, before and after iterations is depicted in fig.12.

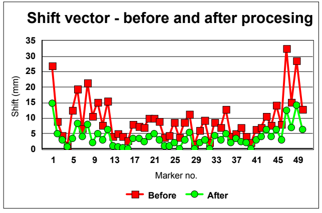


Figure 12. Shift vector improvement

Patch mismatch			
List	Before Ave.shift(mm)	After Ave.shift(mm)	Gain %
Fore-head	15.20	6.19	59.3
Mouth	6.63	2.93	55.8
Cheek	5.60	1.89	66.3
Eyes	6.50	2.44	62.5
Chin	15.19	7.00	53.9

Table 2. Patch mismatch

Patch mismatch (Table.2) is calculated by extracting the patches which were occupied by markers in the relaxed face and that have moved away from those initially mapped markers in the grin action. Patch length, which is calculated as the amount of patch displacement from the marker position, is determined for each patch by searching its neighbors. If the marker is found in one of the immediate neighbors, patch length is set to 1. This procedure is repeated for all the markers in each sub-mesh. It has been noted that the patch lengths were rarely exceeded 2 patches for measured samples of 5 subjects with markers. The average shift computes the average in all shift vectors those belong to the current sub-mesh. With 10 iterations, we were able to reduce this shift error more than 50%.

6. Discussion

Our mesh adaptation method presented here approximates a N-degree polynomial function using least squares estimation. In practice, to wrap the mesh on a measured 512 x 242, 8-bit resolution face range data takes about 15 seconds on Silicon Graphics O_2 workstations. The variance estimations are quite consistent with corrections made to patch shift in different facial actions. Apart from very subtle features, such as frills around the mouth corners, we were able to estimate the facial deformations effectively.

7. Reference

[1] F. Parke, “Parameterized model for facial animation”, IEEE Computer Graphics and Applications, vol. 2(9), 1992, pp. 61-68.

[2] M. Oka, K. Tsutsui, A. Ohba, Y. Kurauchi, T. Tago, “Real time manipulation of texture mapped surfaces”, Computer Graphics, SIGGRAPH, 1987, pp. 181-188.

[3] T. Kurihara, K. Arai, “A transformation method for modeling and animation of modeling and animation of human face from photographs”, State of art in computer animation, Springer-Verlag, 1991, pp. 45-57.

[4] M. Nahas, H. Hutric, M. Rioux, J. Domey, “Facial image synthesis using skin texture recording”, Visual Computer, vol. 6(6), 1990, pp. 337-343.

[5] D. DeCarlo, D. Metaxas, “The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation”, Proc. IEEE Computer Society, 1996, pp. 231-238.

[6] Y. Lee, D. Terzopoulos, K. Waters, “Realistic modeling for Facial Animation”, Computer Graphics Proceedings, SIGGRAPH, 1995, pp. 55-62.

[7]. K. Waters, D. Terzopoulos, “Modeling and animating faces using scanned data”, Visualization and Computer Animation, vol.2, 1991, pp. 123-128.

[8] K. Hasegawa, K. Hattori, Y. Sato, “A High Speed Face Measurement System”, Vision Interface ’99, 1999, pp. 196-202.