ACCV2002: The 5th Asian Conference on Computer Vision, 23--25 January 2002, Melbourne, Australia

Automatic Feature Detection and Pose Recovery for Faces

Alper Yilmaz Mubarak A. Shah
School of Computer Science
University of Central Florida
Orlando, FL-32816 USA
{yilmaz,shah}@cs.ucf.edu

Abstract

In this paper, we propose an approach for detecting facial features and recovering pose in presence of high out of plane rotations for both still images and video streams. To detect the correct features, we assign a confidence number to combinations of feature candidates given the edge map of the face. Feature candidates are determined using probability distribution of color space of skin, eyes and eyebrows. To increase the accuracy of feature detection for video streams, we incorporate motion history information for individual features by weighing the confidence measure according to potential regions of features. Once the best feature combination is obtained, we recover the pose using the centroid of the features assuming orthographic projection.

We conducted experiments on both still images and eleven video sequences including two CNN interviews. In most of the cases, the system performed very well and correctly determined the pose.

1. Introduction

Extracting face features and recovering pose are two challenging problems in computer vision, which have been widely explored by researchers. Many high-level vision applications such as video telephony, face recognition, facial animation, facial feature tracking and MPEG-4 coding, require feature extraction and pose recovery, which is currently done manually or semiautomatically for limited orientations such as frontal views.

In videophones, information of the subject's face, i.e. face orientation and facial expressions, is required for achieving high compression ratio. Similarly, creating realistic facial animation from real images requires initial pose estimation and feature detection of the input face photos for conforming generic wire frame structure to images [6]. Also, most face recognition schemes rely on salient features such as eyes, eyebrows and nose, and relationships between them in 2D without recovering orientation [1].

Given a face image, partial features can be determined in several ways: using shape information [4], motion information, for example determination of eyes using eye blinking [2], and template matching either in image space or eigenspace [3]. All the mentioned approaches have limitations due to high out of plane rotations of the face. However, in non-cooperative environments, it is almost impossible to obtain frontal views. Manjunath et al. [1], proposed a method to feature detection for face recognition in quasi-frontal views by obtaining features using Gabor filters; however their features are not stable, because a feature in one face may not be present in another face, which limits its applicability to pose recovery. Pentland et al. [3] used eigenspace decomposition for individual features, i.e. eyes and mouth. Their method was also affected by rotations and illumination variation.

Even though the features are reliably detected, recovering pose is itself a difficult problem, because it involves feature correspondence between two images and solving the non-linear problem for computing orientation. For recovering pose parameters, Szeliski et al. [6] used manually selected 13 points and solved system of nonlinear equations of high degrees by linearizing.

Another approach uses weak-perspective model, which is an orthographic projection plus scaling, by solving biquadratic equation with three points [5]. Weak perspective model approximates perspective projection by assuming all points on a 3D object are roughly the same distance from camera. Compared to previous models, which are based on algebraic constraints, Alter's approach is motivated geometrically. Similarly, Xu and Sugimoto [8] also used weak perspective model to obtain 3D motion for constructing the 3D structure of the face. Their approach required small transformation.

In this paper, we propose a new approach to detect facial features –such as eyes, eyebrows and mouth– and recover the pose. The rest of this paper is organized as follows. In the next section, we will describe the details of the face feature detection algorithm. In section 3, a pose recovery algorithm, which uses the features found by the proposed feature detection method, will be detailed. Then we will describe the complete algorithm to recover the pose of a face given an image or an image sequence. In section 5, experimental results will be demonstrated to corroborate the proposed method. Finally, we will derive conclusions in sections 6.

2. View Independent Feature Extraction

The feature detection method presented in this paper is based on partial features and it makes use of local relationships between oriented features. In our approach, we utilize the relationships between the local features due to the geometrical structure of the human face. A typical face contains four main parts: eyebrows, eyes, mouth and nose. Among these, nose is the hardest to identify due to low contrast of the skin. Thus, we will concentrate on the eyes, eyebrows and mouth.

Our method for extracting features is divided into two steps. The first step detects the face and extracts the feature candidates. The second step assigns likelihood measure to the geometrical combination of the features. In the next sections, we will present the details of these steps.

2.1. Face Detection and Feature Candidate Extraction

Given an image, most face processing systems require detection of the face. In our framework, we use skin color predicates to obtain the largest uniform elliptic region that corresponds to face. A training set of ten individuals is used to construct a 64-bin RGB lookup table using Kender's [7] method, and then each pixel is labeled as skin or non-skin by looking at the lookup table according to its RGB value. Once the face is detected, we apply morphological dilation to fill the holes and generate a facemask. After mask is generated, next step is to obtain the face feature candidates. In Figure 1, sample input image (a), pixels labeled as skin after skin detection (b) and corresponding facemask (c) are shown respectively.

Facial features, i.e. eyes and eyebrows, can be identified by their contrast with the skin color. The contrast can be utilized by employing different approaches: edge maps, active contour models or eigentemplates. Edge maps are noisy representations and give many false positives depending on the orientation of the face and illumination. Active contour models require rough estimates of locations of individual features [4] and this estimation is not feasible in presence of out of plane rotations. Multi-scale template matching using eigentemplates for individual features has also problems due to illumination variation and shape variations for different orientations.

2.2. Feature Detection

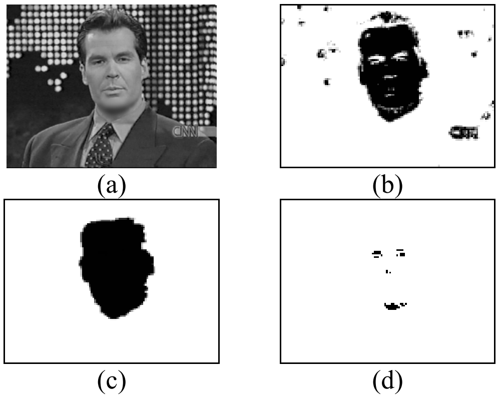

(a) (b)
(c) (d)

Figure 1: Steps in determining the facial feature candidates. (a) Input face image, (b) output of skin detection, (c) face mask obtained by dilation operation on (b), (d) resulting feature candidates.

Given training templates for eyes and eyebrows, we calculate the probability of the color u in each training template by employing a convex, monotonic decreasing kernel profile k, which assigns smaller weights to the locations that are farther from the center of the facial feature. Weighting the probability distribution of color according to the distance from the center increases the robustness of tracking the features in temporal domain, since pixels close to the boundary are likely to be occluded. We use Epanechnikov kernel profile,

$$k_E(x) = \begin{cases} \frac{1}{2} c_d^{-1}(d+2)(1-x) & \text{if } x < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

where d is the number of dimensions and c_d is the volume of d-dimensional sphere, to generate the probability distribution of individual feature since it yields minimum mean integrated square error.

Once we obtain the distributions, for eye and eyebrow training models, we select mean of distributions of all eye templates as eye model and mean of all distributions of eyebrows as eyebrow model.

Given the facemask, we search for possible candidates for individual features (left and right eyes, and eyebrows) that minimize the distance

$$d = \sqrt{1 - \sqrt{\mathbf{p}(\mathbf{x})^T \mathbf{q}}} \qquad (2)$$

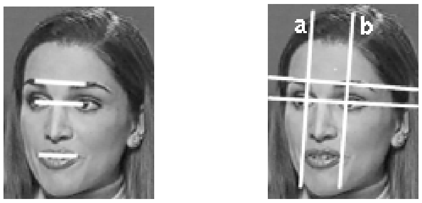
between the model and the candidate location centered at $\mathbf{x}$. In equation 2, $\mathbf{p}(\mathbf{x})$ corresponds to candidate color probability distribution, where kernel centered around $\mathbf{x}$ and $\mathbf{q}$ is the model color probability distribution. Minimizing this distance corresponds to maximizing the likelihood of two distributions, so instead of calculating this distance we consider maximizing the $\mathbf{p}(\mathbf{x})^T\mathbf{q}$ multiplication which in fact corresponds to the cosine of the angle between these vectors.

In the next section, we present a novel approach to determine the correct features by assigning a likelihood measure to different combinations of feature candidates.

2.2. Likelihood

Face has a symmetrical and well-defined structure. As shown in Figure 2, white lines connecting eyes, eyebrows and corners of mouth are parallel, and the mouth always lies between lines a and b, which are perpendicular to the lines connecting eye or eyebrow.

To utilize the geometrical structure given in Figure 2, we used edge map of the face image obtained by Sobel's edge detector, such that, in the ideal case, mouth will be represented by a straight line connecting lines a and b of Figure 2b.



(a) (b)

Figure 2: Geometrical structure of the face (a) eye and eyebrow connecting lines are parallel to orientation of the mouth, (b) mouth resides between lines a and b, which are perpendicular to eye and eyebrow connection lines.

For features found by using the approach given in section 2.1, proposed algorithm calculates the confidence of mouth for each candidate pair given edge map using,

$$C_i(mouth \mid edge, A, B) = \frac{1}{1 + e^{-\frac{x_i - |AB|/2}{k}}} \tag{3}$$

where A and B are feature candidates, $|AB|$ is the Euclidean distance between the feature candidates, k=$|AB|$/10 and x_i is the number of edges on a scan line i. Equation 3 essentially assigns high confidence to a scan line, which has the largest number of edge points. Using Equation 3 for calculating confidence is better than simply using number of edges due to presence of noisy edges. Figure 3 shows the plot of the confidence measure as a function of number of edges on a scan line for $|AB|$=30. In this figure, the vertical axis refers to the value calculated using Equation 3 and horizontal axis is the number of edge points on a scan line.

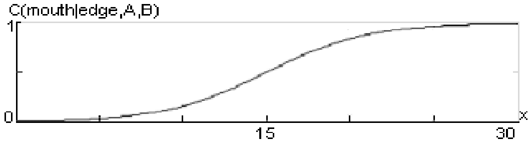


Figure 3: Plot of C(mouth|edge,A,B) as a function of number of edges for |AB|=30.

Given the reference image in Figure 4a, for each pair of feature candidates (AB, AC, AD, CB, CD…), the algorithm scans the edge map in orientation parallel to line connecting the feature candidates and calculates the confidence C_i of encountering mouth on i^{th} scan line, and finds the likelihood of j^{th} feature pair by

$$L_j = \arg\max_{i \in \Gamma_j} C_i(mouth \mid edge, A, B) \tag{4}$$

where Γ_j is scan line space for the j^{th} feature pair. Once feature likelihood is calculated for all pairs, we select the best feature combination, eyes or eyebrows and mouth, by

$$\varphi = \arg\max_{t \in \Psi} L_t \tag{5}$$

where Ψ is the space for all possible pairs of feature candidates and φ is the best feature pair. Equation 5 in conjunction with Equation 4 also gives the mouth location, which will be denoted by M. Though, φ gives the best possible pair, this pair can be either eyes or eyebrows, because we have not imposed any constraints on the solution given in Equations 3-5.

In Figure 4a, A, B is the features pair in φ, AB is the line connecting A and B and a and b are perpendicular lines. Figure 4b shows the plot of confidence for pair A and B, which is calculated using Equation 3 for the reference image shown in Figure 4a.

The next phase of the algorithm uses A and B and looks for other possible pair C and D whose confidence were already evaluated in the previous step; the algorithm makes its decision based on the following criteria,

$$\phi = \arg_{m,t \in \Gamma \wedge m \neq t | AB \in t, CD \in m} \begin{pmatrix} (|AB| \approx |CD|) \wedge \\ (AB // CD) \wedge \\ (dist(CD, AB) < \varepsilon) \end{pmatrix} \tag{6}$$

where |.| denotes length of vector, // denotes parallel lines, *dist* denotes distance between lines and $\wedge$ denotes logical and operation. Let A, B correspond to correct pair of features, Equation 6 essentially locates the other pair C, D such that length of AB is similar to length of CD and the line connecting A, B is parallel to the line connecting C, D and the distance between these to lines is smaller than a predefined threshold.

If no features are found by this operation then algorithm looks at the distance from the remaining candidates to one of the perpendicular lines of AB (lines a or b, Figure 2b). This distance is given by

$$D_i = \frac{1}{|AB|}[-(A_y - B_y)A_y - (A_x - B_x)A_x + (A_y - B_y)y_i + (A_x - B_x)x_i] \tag{7}$$

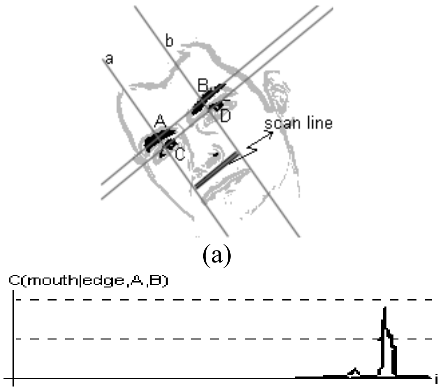
where $|AB|$ is the length of AB, x_i and y_i are centroid locations of ith candidate and $i \in \Psi$-$\{A,B\}$. Note that in Equation 7, D_i is only dependent on terms including x_i and y_i, so the algorithm selects feature candidate by

$$\delta = \arg\min_{i \in \Psi - \{A,B\}} \left((A_y - B_y)y_i + (A_x - B_x)x_i\right) \tag{8}$$

where δ is the selected feature candidate and estimates the corresponding feature according to the criteria given in Equation 6. Once two pairs are found, we will label the features pairs as eyes or eyebrows using,

$$\begin{aligned} MD_{AB} &= (A_x - B_x)M_x - (A_y - B_y)M_y \\ MD_{CD} &= (C_x - D_x)M_x - (C_y - D_y)M_y \end{aligned} \tag{9}$$

where MD denotes distance of mouth to lines AB and CD. If MD_{AB}<MD_{CD} then A, B will be labeled as left and right eye, and C, D will be labeled as left and right eyebrow;



(a)

(b)

Figure 4: (a) Reference image for calculating the confidence, (b) Plot of confidence for pair A and B, which calculated using Equation 3 for each scan line of reference (a).

otherwise *AB* and *CD* will switch labels.

In the next section, we will present a novel approach to recover the pose of the face by using the feature locations found by the proposed algorithm.

2.4. Imposing Video Constraints

In the previous section, we used information obtained using a single image. In video, we have a sequence of frames, therefore we can use the information obtained in the previous frames for the current frame. Given a local feature in frame *k*, we can increase the likelihood of selecting corresponding feature in frame *k*+1 by generating a potential region for the feature.

To generate the potential region for a feature, e.g. right eye, we accumulated the respective locations of motion history map for right eye for consecutive frames using accumulation filter,

$$R_1(x,y;\lambda)=\frac{1}{2\lambda}e^{-\frac{\sqrt{x^2+y^2}}{\lambda}} \qquad (10)$$

where λ is the standard deviation of the filter. Accumulation filter is given in Figure 5a. In Figure 5b, we show the motion history map for the right eye feature for frame numbers 1, 46, 85 and 136. The gradient regions on the images define potential regions.

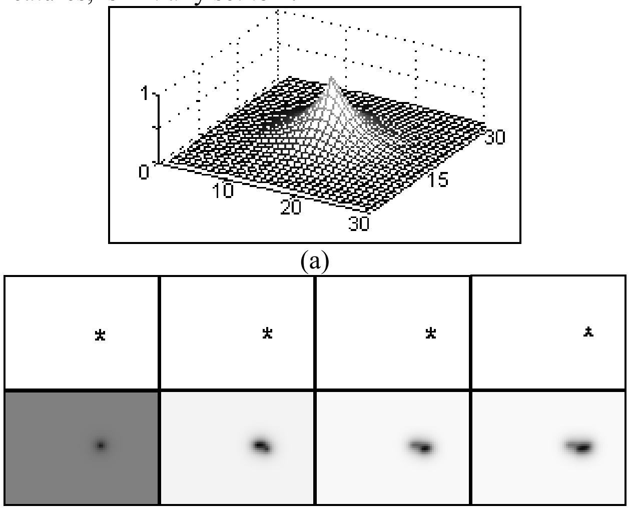
To incorporate motion history in feature detection, we calculate weighted confidence measure, which is a modified version of Equation 3. Weighted confidence measure is defined by

$$C_i(mouth\,|\,edge,A,B)=w_{ABC}\frac{1}{1+e^{-\frac{x_i-\frac{|\mathrm{AB}|}{2}}{k}}} \qquad (11)$$

where w_{ABC} is the weight assigned by the motion history of features *A*, *B* and *C* such that *A* and *B* corresponds to eyes or eyebrows and *C* corresponds to mouth and is given by,

$$\frac{1}{w_{ABC}}=\frac{1}{m_A(x_A,y_A)}+\frac{1}{m_B(x_B,y_B)}+\frac{1}{m_C(x_C,y_C)} \qquad (12)$$

where *x* and *y* is centroid of the features *A*, *B* and *C*. Motion history map, which is saved separately for all features, is initially set to 1.



(a)

(b)

Figure 5: (a) Accumulation filter for potential region generation, (b) first row spatial location of right eye, second row corresponding motion history map obtained by accumulating the respective locations using accumulation filter for frames 1, 46, 85 and 136 from left to right.

To efficiently use the motion history of features, we extended the system to handle cases where no feature locations are detected due to color predicates. For the frames where feature detection failed, we estimate the new locations by looking at the previous displacements of the features.

3. Recovering Pose

Linearly separable three points in camera coordinates define a triangle in the image plane. The orientation of the plane is proportional to the change of distance between points in image coordinates. Given no additional information about the depth of these points and the extrinsic camera parameters, it is feasible to assume orthographic projection to simplify the problem for forcing real time computation, since solving for depth under perspective projection yields a non-linear system.

Features obtained by the algorithm outlined in the preceding section constitute basis to form the triangle in the image plane. We will refer to constellation of these points as T-Structure given in Figure 6, where point *A* is left eyebrow, point *B* is right eyebrow and point *C* is mouth. Note that *A* and *B* can also be selected as left and right eye according to the confidence measures given in Equations 3 and 11.

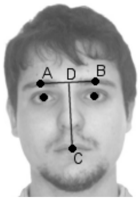


Figure 6: T-structure superimposed on a face image.

To recover the pose of the face, our algorithm uses two steps; first we find the rotation of the face and then we find the translation. Given Figure 6, let three points *A*, *B* and *C* have the same depth z=Z_{ABC}, i.e. frontal view, and their rotated versions be denoted by *A''*, *B''* and *C''*. Since we are interested in calculating the rotation, we will translate T-structure such that *D* resides on the origin and Z_{ABC}=0, and recover rotations around *y*-axis, *z*-axis and *x*-axis in the stated order. Rotation of the feature points *A* is,

$$\begin{pmatrix} A'''_x \\ A'''_y \\ A'''_z \end{pmatrix} = R_\alpha R_\gamma R_\beta \begin{pmatrix} X_{AD} \\ 0 \\ Z_{ABC} \end{pmatrix} \tag{13}$$

where A''' is obtained by rotating *A* around *y*-axis by β, *z*-axis by γ and *x*-axis by α and $A = (X_{AD}, 0, Z_{ABC})^T$. Similarly, we can find B''' and C''' given $B = (X_{BD}, 0, Z_{ABC})^T$ and $C = (0, Y_{CD}, Z_{ABC})^T$, where $X_{AD} = -X_{BD}$ due to symmetrical structure of the face. Equation 13 along with B''' and C''' constitutes a linear system of equations, and the solutions of γ and β are given

$$\gamma = \arcsin \frac{C'''_x}{C_y} \tag{14}$$

$$\beta = \arccos(A''' / A_x \cos\gamma) \tag{15}$$

For solving α we will use

$$\begin{aligned} A'''_y &= (A_x \cos\beta \sin\gamma)\cos\alpha - (A_x \sin\beta)\sin\alpha \\ C'''_y &= (C_y \cos\gamma)\cos\alpha \end{aligned} \tag{16}$$

derived from the system of Equation 13. Solving α yields,

$$\alpha = \arccos \frac{C'''_y}{C_y \cos\gamma} \tag{17}$$

$$\alpha = \arcsin \frac{A_x C'''_y \cos\beta - C_y A'''_y \cos\gamma}{A_x C_y \cos\gamma \sin\beta} \tag{18}$$

Rotation around *x*-axis can be recovered using either Equation 17 or 18; we used Equation 17 because Equation 18 introduces more floating point operations and more prone to quantization errors. Note that the approach uses projections of *A*, *B* and *C* points for a scaled generic frontal view face obtained ahead of time. After recovering the rotation angles, we can obtain the translation parameters simply by considering the displacement of the points.

4. Complete Algorithm

A complete summary of the algorithm is as follows:

1. Generate eye and eyebrow model using a set of training templates (we used 20 templates for eye and eyebrow).
2. Given an arbitrary image, detect the face using skin predicates obtained from a training set of ten individuals.
3. Perform morphological dilation and generate facemask.
4. Locate the feature candidates in the face region using eye and eyebrow model.
5. Obtain edge map of face image using Sobel's detector.
6. Calculate likelihood of feature candidate combinations by applying Equations 3-6.
7. Using three feature points generated in step 6, calculate rotations by Equations 14, 15 and 17 and obtain translation by considering the displacements of the rotated points.

In the next section, we will present the experiments and discussions about feature detection and pose recovery algorithms outlined above.

5. Results and Discussions

In order to show the efficiency of the our approaches for face feature detection and pose recovery, we used three different sets, one for training the system for generating color predicates of skin and eyes/eyebrows, and two for testing. The training set is composed of ten individuals with frontal views. The test set is composed of still face images of five individuals with five different poses, CNN interview video streams with 150 frames each and nine video sequences, which were shot in computer vision laboratory. Still image test set and vision lab streams encompass variety of poses. Interview streams have limited poses, however, the lighting conditions differ from the other sets.

To demonstrate the accuracy of feature detection algorithm, we used all still images and the frames of the video streams. For the video frames, we treat every frame separate from the others, such that we didn't include any motion information and used Equation 3 instead of Equation 11. Visually looking at the detections, we determine that our approach was able to correctly identify features in 1018 out of 1358 images. Figure 7, shows results on some of the test set face images.

As seen in Figure 7a, proposed method for still images correctly detected the feature locations. In Figure 7b, since the eyebrow colors are weak to be detected only eyes are detected. However, in Figure 7c, due to noisy edge structure, the system failed to detect the correct feature locations. Our analysis showed that the algorithm performed poorly for two cases: high rotations in presence of beard and CNN sequences when the interviewee closed their eyes. For the CNN sequences the system couldn't detect eyebrows due to light colors.

To see the performance of the system for video sequences, we used motion history map discussed in section 2.4. The performance of the system, improved drastically and most of the wrong feature detections similar to results shown in Figure 7d were correctly localized. Some of these results are shown in Figure 8.

For measuring the accuracy of the pose recovery algorithm, we visualized the recovered rotations and translations by fitting Candide wire frame face model, composed of 108 triangles, to the input face image. First, we rotate and translate the generic mesh and then we conform the mesh using 5 feature points on both input face image and generic mesh (note that, in some cases only 3 of features are available and the affine transformation requires at least 3 points). The pose recovery method highly depends on the locations of features detected by the feature detection scheme. In all cases, where the facial features were correctly localized, pose recovery algorithm gave correct estimates of the pose. In other cases, where the features were incorrectly detected our system gave just a rough approximation of the pose.

In Figure 9, we show the mesh superimposed on the face images from test set after recovering the pose; recovered rotation angles (α,β,γ) from left to right are (-11.48, -16.26, 21.10), (25.03, -25.12, 12.29), (16.26, -19.95, 47.73), (0.00, 67.67, 5.74), (0.00, -52.41, -2.87).

6. Conclusion

We have proposed a new approach for detecting facial features, i.e. eyes, eyebrows and mouth from color images. Our feature detection method uses a weighted confidence measure to select best combinations of feature candidates.

The experiments demonstrate the success of the framework in presence of large out of plane rotations. We increased the accuracy of the approach for video streams by using motion history information.

One potential application of our work is in videophones. In this context, the transmitter will only send the transformation parameters composed of 5 numbers, corresponding to 3 rotation angles and 2 translations and the receiver will be able to synthesize video sequence using the wire frame model, transformation parameters and the texture map.

10. References

[1] B.S. Manjunath, R., C. Malsburg, “A feature Based Approach to Face Recognition,” *Proc of CVPR*, 1992, pp. 373-378.

[2] L.P. Bala, K. Talmi, J. Liu, “Automatic Tracking of Faces and Facial Features in Video Sequences,” *Proc of Picture Coding Symp,* 1997, Berlin, Germany.

[3] B. Moghaddam,A. Pentland, “Probabilistic Visual Learning for Object Representation,” in S. K. Nayar, T. Poggio (edts), *Early Visual Learning*, Oxford Unv. Press, New York, 1996.

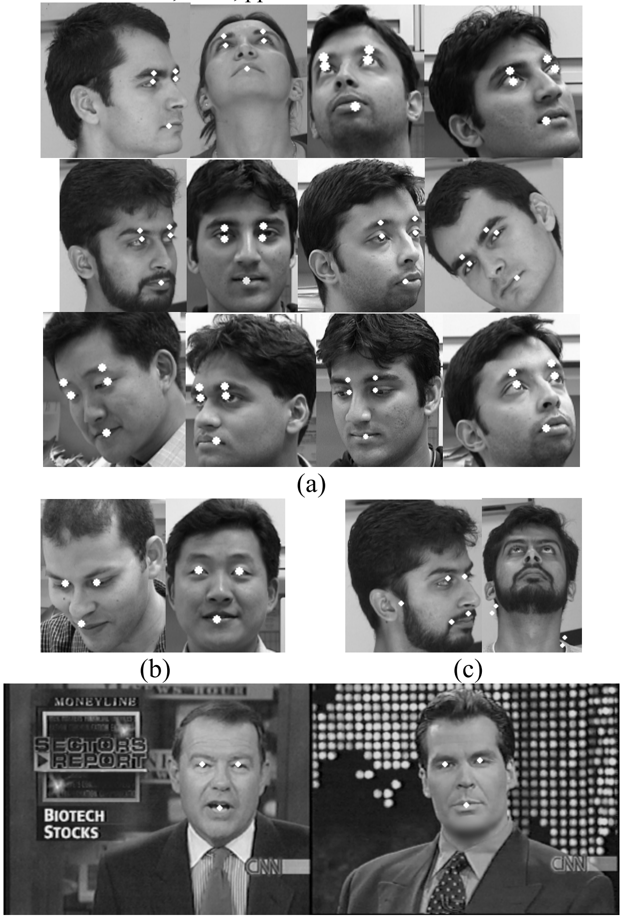
[4] M. Kass, A. Witkin and D. Terzoupoulos, “Snakes: Active Contour Models,” *Int. Jour. of Computer Vision*, 1998, pp. 321.

[5] T.D. Alter, “3-D Pose from 3 Points Using Weak Perspective,” *IEEE Trn. on PAMI,* Vol.16, No.8, pp. 802.

[6] F. Pighin, J. Hecker, D. Lichinski, R. Szeliski, D. H. Salesin, “Synthesizing Realistic Facial Expressions from Photographs,” *Proc. SIGGRAPH,* 1998, pp. 75-84.

[7] R. Kjeldsen, J. Kender, “Finding Skin in Color Images,” Proc. of the 2^{nd} Int. Conf. on Automatic Face and Gesture Recognition, pp. 312-317.

[8] G. Xu, N. Sugimoto, “A linear algorithm for motion from three weak perspective images using Euler angles,” *IEEE Trans. on PAMI,* Vol. 21, 1999, pp.54-57.



(a)

(b)

(c)

(d)

Figure 7: Detected facial features for still images a, b and c; and video frames without motion information d.

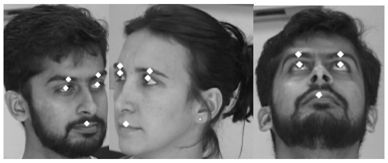


Figure 8: Correctly localized facial features by using motion history map.

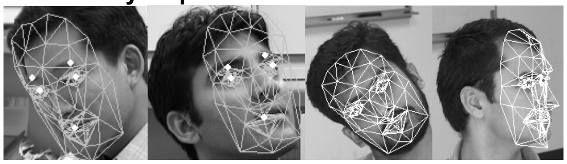


Figure 9: Mesh superimposed on the face image after recovering the face pose under affine transformation.