A Model-based Algorithm For Facial Feature Extraction From a Photograph

Adnan M. Alattar
and
Sarah Rajala

Center for Communications and Signal Processing
Electrical and Computer Engineering Department
North Carolina State University

CCSP-TR-88/26

October, 1988

A Model-Based Algorithm For Facial Feature Extraction From A Photograph

Adnan M. Alattar and Sarah A. Rajala
Center for Communications and Signal Processing
P.O. Box 7911, North Carolina State University
Raleigh, NC 27695-7914

Abstract— *The process of locating facial features such as the eyes, nose, mouth and chin in a head and shoulders photograph is very important to any automatic face recognition system. In this paper, a new algorithm for extracting these major features is developed. The algorithm is based on a model for the human head in a photograph. Since the human head can be viewed as an ellipsoid, the head view in the photograph can be modeled by an ellipse. The algorithm estimates the parameters of the ellipse which best fits the head view in the photograph, and uses these parameters to estimate the locations of the facial features. It then adjusts the vertical coordinates of the mouth and nose by using the signature of the pixels in a vertical slit around the nose and mouth. The algorithm has been implemented, and simulation results have verified the accuracy of the estimation of the facial features' locations.*

1. Introduction

Automatic human face recognition is an important and challenging problem in computer vision. In spite of the importance of this problem to applications such as security and law-enforcement, limited success has been achieved in automatic recognition of faces. Most of the existing techniques are based on either verbal coding or geometrical coding of the image, followed by a sequential or matching algorithm.

While verbal coding describes a picture verbally, geometrical coding uses distance measurements to describe a picture. Goldstein et. al. [1] used verbal coding to develop an interactive face recognition system. Harmon et. al. [2,3,4] used geometrical coding in automatic identification of human face profiles. Each face profile was represented by a vector of seven automatically-extracted fiducial marks. These vectors were then used to compare the profile pictures. Sakai et. al. [5] and Bromley [6] developed algorithms for automatic location of the facial features in front view images. Sakai [5] used the vertical and the horizontal signatures of the pixels in a slit of predetermined dimensions to determine the location of the facial features. The slit moves around the picture in search of the location of the desired feature. This location must be consistent with the locations of other features. Whenever the algorithm detects an inconsistent location, the process is repeated in an iterative manner. Bromely [6] used a similar algorithm, except that the horizontal and vertical signatures of the global image, rather than of a slit, was used.

The location of facial features is necessary not only in face identification, but also in constructing composite images from parts of photographs. Gillenson and Chandrasekaran [7] developed a computer system called *Whatsisface*, which utilizes a database of human facial features to create facial images on a CRT. Wiederhold [8] developed a computer system to simulate the *Menolita Montage Synthesizer*, which blends features from different pictures to produce a composite for criminal identification. In both *Whatsisface* and the *Menolita Synthesizer*, the location of the feature to be extracted or projected is determined manually. Recently, Alattar and Rajala [9] developed a new image coding technique in which facial features are extracted from the input image and then matched to a database. The order of the features in the database, rather than the features themselves, is transmitted to the decoder, where the features are then extracted from an identical database and used to construct a composite image.

In this paper, a one-pass algorithm is developed which quickly and accurately locates the desired facial features. The new extraction algorithm is a model-based algorithm. The art of drawing the human head is used to develop a model for the head in the photograph. In general, this model is good for all types of pictures; however, our discussion and implementation will be restricted to photographs of people with short hair. To extend the work to include people with long hair, one needs to model the face instead of the head. The paper is organized into four sections. Section (2) describes the head model which is used in extracting the necessary features. Section (3) describes an algorithm for feature extraction. Section (4) presents and discusses the simulation results. The last section contains the conclusion and further remarks.

2. Model for Head and Shoulder Images

The head of a human being can be thought of as an ellipsoid that sits on the top of the torso and is attached to it at the center by the neck. The head movement can then be viewed as rotations α, β , and θ around the x, y, and z axes, respectively, which pass through the center of the ellipsoid (see Fig. 1). Under this ellipsoid model, the head view in a photograph has the shape of an ellipse. The elongation and orientation of this ellipse are related to the head rotation angles α, β and θ. For example, in the front view position, where the angles α and β are both zero, the length of the minor axis is about two thirds the length of the major axis [10,11]. However, the ratio changes as the head moves up or down, and therefore is related to the value of angle α (assuming that the x-y plane is the image plane). The symmetry of the head view in the original image is related to the angle β, and the tilt of the head, i.e., the orientation of the minor axis of the ellipse, is indicated by the angle θ.

It is well known in the art of drawing the human head that the head is approximately five eye-lengths wide [10,11] (see Fig. 2). Both eyes lie on the midway line and the distance between them is equal to almost one eye-length. The nose starts at the center of the face and descends to a point mid-way between the bridge of the nose and the base of the chin. The width of the nose at its base is also equal to approximately one eye-length. The mouth barrel starts at the nose and extends two thirds of the distance down from the nose to the chin. The sides of the barrel align with the centers of the eye sockets. This information is used to locate the facial features with respect to the center of the ellipse which models the human head in a photograph. The locations are then adjusted to obtain more accurate results.

Let (c_x, c_y) be the center of the ellipse that fits the head contour. Also let l_x, l_y and θ denote the minor and major semi-axes and the orientation of the ellipse, respectively. Using the previously described model of the head and the geometry of the ellipse, the locations of the eyes, nose, and mouth in terms of (c_x, c_y), θ, l_x, and l_y are given below.

Eyes:

If the length of each eye is e_l and the centers of the right and left eyes are (cre_x, cre_y) and (cle_x, cle_y), respectively, then,

$$e_l = \frac{2l_x}{5} \tag{1a}$$

$$\begin{aligned} cre_x &= cx + e_l\cos\theta \\ cre_y &= cy + e_l\sin\theta \end{aligned} \tag{1b}$$

$$\begin{aligned} cle_x &= cx - e_l\cos\theta \\ cle_y &= cy - e_l\sin\theta \end{aligned} \tag{1c}$$

Nose:

If the center, length, and base width of the nose are denoted by (nc_x, nc_y), n_h and n_w, respectively, then,

$$nc_x = c_x - \frac{l_y}{4}\sin\theta \tag{2a}$$

$$nc_y = c_y + \frac{l_y}{4}\cos\theta$$

$$n_h = \frac{l_y}{2} \tag{2b}$$

$$n_w = \frac{2l_x}{5} \tag{2c}$$

Mouth:

If the center, height and width of the mouth barrel are denoted by (mc_x, mc_y), m_h and m_w, respectively, then,

$$mc_x = c_x - \frac{2l_y}{3}\sin\theta \tag{3a}$$

$$mc_y = c_y + \frac{2l_y}{3}\cos\theta$$

$$m_h = \frac{l_y}{3} \tag{3b}$$

$$m_w = \frac{4l_x}{5} \tag{3c}$$

Equations (1)-(3) are used in the facial feature extraction algorithm to estimate the location of the facial features.

The parameters of the ellipse are estimated by fitting the head contour points with the ellipse function,

$$f(x,y;a,b,c,d,e) = ax^2 + 2bxy + cy^2 + dx + ey - 1 = 0 \tag{4}$$

It is easy to show that the center (c_x, c_y) of this ellipse is given by

$$c_x = \frac{be - cd}{2(ac - b^2)} \tag{5a}$$

$$c_y = \frac{bd - ae}{2(ac - b^2)} \tag{5b}$$

and the orientation θ is given by

$$\theta = \tfrac{1}{2}\tan^{-1}\left(\frac{b}{a - c}\right) \tag{5c}$$

The major and minor semi-axes, l_x and l_y, are given by

$$l_x = \left[\frac{(c-a)(1 - ac_x{}^2 - bc_x c_y - cc_y{}^2 - dc_x - ec_y)}{(ac - a^2 - 2b^2)\sin^2\theta + (c^2 - ac + 2b^2)\cos^2\theta}\right]^{1/2} \tag{5d}$$

$$l_y = \left[\frac{(c-a)(1 - ac_x{}^2 - bc_x c_y - cc_y{}^2 - dc_x - ec_y)}{(ac - a^2 - 2b^2)\cos^2\theta + (c^2 - ac + 2b^2)\sin^2\theta}\right]^{1/2} \tag{5e}$$

A simple way to fit the head contour with the ellipse function is to minimize the fitting error, which is defined by

$$S = \sum_{i=1}^{n} (\Delta x_i)^2 + (\Delta y_i)^2 \tag{6}$$

where Δx_i and Δy_i are the x and y deviations of the head contour point (x_i, y_i) from the ellipse contour. An iterative solution for this problem using the Lagrange multiplier method is shown in the Appendix. The algorithm starts by guessing the parameters of the ellipse. This initial guess can be some predefined values or the values of the parameters of the ellipse that fits five head contour points which are almost equidistant from each other. The algorithm then updates the initial guess by solving a set of linear equations for the deviation from the initial guess. The process continues until the deviation is so small that no significant improvement in the estimated parameters can be obtained with further iterations.

3. Facial Features Extraction Algorithm

In this section, an algorithm for extracting the facial features, based on the model developed in section (2), is described. The algorithm is summarized in the flowchart of Fig. 3. It consists of five consecutive stages. The first four stages are for the parameters estimation of the ellipse which best fits the head contour. These stages are edge detection, thinning, extraction, and curve fitting. An edge operator is used to enhance the edges. Then, the enhanced image is thresholded to separate significant edges from the background. These edges may be neither thin nor continuous. Although edge continuity is not necessary, it is desired, since it facilitates the extraction of the head contours. Furthermore, thin edges are needed to reduce the ellipse fitting error which the fitting algorithm tries to minimize. Hence, after thresholding the enhanced image, a thinning operator is needed to produce the thinned edges. The outer most head contours are extracted from the thinned image, and the parameters of the ellipse that best fits the contour points are estimated. From the parameters of the ellipse the location of the facial features are estimated.

The fifth stage is the final adjustment of the locations of the facial features. It is necessary because the location of the nose or the mouth with respect to the center of the ellipse varies slightly from one person to another. This stage begins with calculating the vertical signature of the pixels in a slit around the estimated nose or mouth location. These pixel values are taken from the original binary image before thinning. The base of the nose or the top and the bottom of the mouth are found from the signature by marking the start and end of the non-zero values in the slit.

The details of the above process can be summarized in the following algorithm:

Edge Detection:

(1) *Given a head and shoulders image, convolve the image with Sobel's [12] templates (see Fig. 4a) to compute the magnitude of the image gradient. Threshold the image globally to produce a binary edge image.*

Edge Thinning:

(2) *Remove all pixels in the image that match one of the Chen et. al [14] templates (a)-(h), but leave pixels that match template (i) or (j) (see Fig. 4b). Repeat this step until no further changes occur.*

Separating the Head from the Rest of the Body:

(3) *Define the boundary which separates the head in the picture from the rest of the body. The boundary is positioned at $\sim$ 5 pixels above the shoulders (first non-zero pixel encountered vertically). Set all pixels in the rows below the boundary to zero values.*

Outermost Head Contour Extraction:

(4) *Scan the image horizontally from left to right and bottom to top, until a non-zero pixel x_0 is encountered.*

(5) *Inspect the pixels in an* $M \times M$ *window (initially* 3×3*) around the current non-zero pixel* x_i*. If another non-zero pixels* x_{i+1} *is found, go to step (6), otherwise, increase M and repeat this step as long as M is less than a certain threshold. Whenever M is greater than the threshold go to step (7).*

(6) If M is 3 save the location of x_i *and make the value of pixel* x_i *zero. If M is greater than 3, save the location of* x_i *and the locations of all pixels on a straight line between* x_i *and* x_{i+1}*, then make the pixel value of pixel* x_i *zero. Make pixel* x_{i+1} *the current pixel and go to step (5).*

Approximating the Chin Contour:

(7) *If the last non-zero pixel on the contour is not the same as the first, connect them with an arc or two line segments.*

Estimating the Parameters of the Ellipse:

(8) *Initialize the parameters* a, b, c, d, and e.

(9) *Use the contour points to compute* p_i *in Eq. (a.6), and solve Eq. (A.4) for* Δa, Δb, Δc, Δd, *and* Δe, *(see the Appendix).*

(10) *Update the parameters* a, b, c, d and e *using the deltas from step (9). If the magnitude of the deltas is not small enough, go to step (9); otherwise go to step (11).*

Estimating the locations of the Facial Features:

(11) *Use the parameters from step (10) to estimate the locations of the facial features as given by Eq. (1)-(3).*

Adjusting the locations of the Facial Features:

(12) *Compute the vertical signature in a* $(mw \times l_y)$ *slit centered at* $(c_x, c_y + ½l_y)$*; then choose the position of the last non-zero value in the slit as the base of the mouth.*

(13) *Search the signature for the valley of zero values that is closest to the center of the slit. Choose the starting point of this value as the base of the nose and the end point as the top of the mouth.*

(14) *Use the top and base of the mouth to compute the vertical coordinate of the center of the mouth. Use the base of the nose and the center of the ellipse to compute the vertical coordinate of the center of the nose.*

4. Implementation and Simulation Results

The above facial features extraction algorithm was successfully implemented in the *C* language. The Sobel edge operator [12] was used to enhance the edges in the image. Sobel templates are shown in Fig. 4a. Using the histogram of the enhanced image, the value of the global threshold was determined. This value was used to separate the edges from the background and to convert the image into a binary image. The thinning algorithm of Chen et. al. [14] was used to produce thin edges.

The Chen et. al. algorithm is an iterative parallel algorithm that has proven to be faster than other thinning algorithms due to its property of true parallelism. The templates of this algorithm are shown in Fig. 4b.

When this algorithm was applied to the image of Fig. 5, the images of Fig. 6 were obtained. Fig. 6a is the edge-enhanced image produced by the Sobel operator. The edges in this image were enhanced more than those obtained by other operators. For instance, they are stronger and thinner than those produced by Roberts and Davis operators [13]. Also, the Sobel operator outperformed Laplacian operator, which produced a lot of undesired noise and inaccurate edges [13]. When the average Laplacian operator suggested by Sakai et. al. [5] was used, thick and inaccurate edges were obtained. This type of edge is undesirable because of the later thinning process. The intensity values of the pixels in the enhanced-edge image were rescaled to gray levels (0-255). To produce a binary edge image, each intensity value was thresholded at a global threshold of 35, which was determined from the histogram of the enhanced image. The thresholded image is shown in Fig. 6b. Finally four iterations of the Chen et. al. thinning algorithm produced the thinned image of Fig. 6c.

Since the inside contours of the face are of no significance to the contour extraction process, they were first deleted from the image to avoid confusion with the head contours. In spite of the discontinuity of the edges, the previously described contour tracking algorithm has successfully extracted the outer most edges of the head, as shown in Fig. 6d. The curve-fitting algorithm used the contours of Fig. 6d to produce the ellipse of Fig. 6e. The estimated ellipse was superimposed over the head contour for comparison. It was a good fit; the major inaccuracy of the estimation resulted from the fact that the chin part of the contour was missing (Fig. 6d), and it had to be approximated by two straight line segments. The curve-fitting algorithm was fast, and it took only four iterations to obtain an accurate estimation of the ellipse parameters.

Next, the ellipse parameters were used to estimate the locations of the major features of the face, and the locations indicated by crosses on the image of Fig. 7 were obtained. Notice the shift in the mouth and lower part of the nose. This shift has two reasons. The first is the approximation of the lower part of the head by two straight lines. If a curve segment had been used for this approximation, a better approximation of the vertical coordinates of the mouth and the nose would have been obtained. The second reason is that the length of the lower half of the face relative to the upper half varies significantly from one person to another. Hence, the vertical coordinates of the mouth and nose were adjusted using the vertical signature of the pixels in a slit centered at the estimated center of the nose, of length equal to the length of the semi-major axis of the ellipse, and of width equal to the width of the mouth. The adjusted locations are shown in Fig. 8. Obviously, the exact locations of the facial features were found without matching. Notice that it is not necessary to adjust the location of the eyes. The reason is that the length and locations of the eyes with respect to the length of the minor axis of the face is almost the same for all people.

This algorithm has also been tested on images taken from a sequence of pictures of a speaker. The motion in this sequence was not restricted to planar motion, but consisted of natural and normal motion. The speaker naturally rotates his head to both sides and nodes up and down as he speaks. Ten pictures of different rotation angles were used in the testing process. The rotation angles of those images varied from $0.27°$ to $7.36°$. The picture in Fig. 5 represents the maximum rotation of $7.36°$. In all cases, the algorithm produced accurate estimations of the three major facial features.

5. Conclusion

A new algorithm has been developed which locates the major features of the face in a head and shoulders photograph. The algorithm is based on a model developed expressly for head and shoulders images. The algorithm has been successfully implemented. Results show that the major facial features can be located in a photograph, in a simple and fast manner. Although the photographs we used were restricted to front view short-haired male head, other types of photographs may be used with slight modifications of the algorithm. For instance, by extending the algorithm to deal with the relations between feature locations and head rotation angles, profile and semi profile photographs can be successfully dealt with. Also, by modeling the face rather than the head in the photograph, the previously described algorithm can be used with pictures of long-haired people.

6. Appendix

The following is the solution for the optimization of the parameters of the ellipse that best-fits the head contour.

Let a, b, c, d, e denote the true parameters of the ellipse of best fit, and let a', b', c', d', e' denote an approximate values of these parameters. If (x_i, y_i) denotes a point that lies on the contour of the head, and (X_i, Y_i) denotes the nearest point on the ellipse, then

$$f(X_i, Y_i; a, b, c, d, e) = 0 \tag{A.1}$$

Where $X_i = x_i - \Delta x_i$, $Y_i = y_i - \Delta y_i$, $a = a' - \Delta a$, $b = b' - \Delta b$, $c = c' - \Delta c$, $d = d' - \Delta d$, $e = e' - \Delta e$. Assuming Δx_i, Δy_i, Δa, Δb, Δc, Δd, and Δe are very small, Eq. (A.1) can be expanded around the point $(x_i, y_i, a', b', c', d', e')$ by a Taylor series expansion. When the higher order terms are ignored, the expansion is given by

$$\begin{aligned} f(x_i, y_i, a', b', c', d', e') &- \frac{\partial f_i}{\partial x}\Delta x_i - \frac{\partial f_i}{\partial y}\Delta y_i - \frac{\partial f_i}{\partial a}\Delta a - \frac{\partial f_i}{\partial b}\Delta b \\ &- \frac{\partial f_i}{\partial c}\Delta c - \frac{\partial f_i}{\partial d}\Delta d - \frac{\partial f_i}{\partial e}\Delta e = 0 \end{aligned} \tag{A.2}$$

for all $(i = 1, 2, \ldots, n)$, where $\frac{\partial f_i}{\partial x}\Delta x_i$ represents the value of $\frac{\partial f}{\partial x}\Delta x$ at $(x_i, y_i, a', b', c', d', e')$.

This problem is a constrained optimization problem in which the performance measure S (see Eq. (6)) is to be minimized under the constraint of Eq. (A.2). The

Lagrange multiplier method can be used to solve this problem. If we denote h_i by the left hand side of Eq. (A.2), the problem is reduced to minimizing the following augmented function

$$g = \frac{1}{2}S - \sum_{i=1}^{n} p_i h_i \tag{A.3}$$

where p_i is the Lagrange multiplier and S is the performance measure. Differentiating this equation with respect to Δx_i, Δy_i, Δa, Δb, Δc, Δd, and Δe, respectively, and setting the result equal to 0, produces the following equations

$$\sum_{i=1}^{n} p_i \frac{\partial f_i}{\partial a} = 0 \tag{A.4a}$$

$$\sum_{i=1}^{n} p_i \frac{\partial f_i}{\partial b} = 0 \tag{A.4b}$$

$$\sum_{i=1}^{n} p_i \frac{\partial f_i}{\partial c} = 0 \tag{A.4c}$$

$$\sum_{i=1}^{n} p_i \frac{\partial f_i}{\partial d} = 0 \tag{A.4d}$$

$$\sum_{i=1}^{n} p_i \frac{\partial f_i}{\partial e} = 0 \tag{A.4e}$$

$$\Delta x_i + p_i \frac{\partial f_i}{\partial x} = 0 \qquad \text{for } (i=1,2,...,n) \tag{A.5a}$$

$$\Delta y_i + p_i \frac{\partial f_i}{\partial y} = 0 \qquad \text{for } (i=1,2,...,n) \tag{A.5b}$$

Substituting Eq. (A.5) back into Eq. (A.2) and solving for p_i yields,

$$p_i = \frac{1}{m_i}(f_{ai}\Delta a + f_{bi}\Delta b + f_{ci}\Delta c + f_{di}\Delta d + f_{ei}\Delta e - f_i) \qquad \text{(A.6)}$$

where

$$m_i = (\frac{\partial f_i}{\partial x})^2 + (\frac{\partial f_i}{\partial y})^2$$

$$f_{ai} = \frac{\partial f_i}{\partial a}$$

$$f_{bi} = \frac{\partial f_i}{\partial b}$$

$$f_{ci} = \frac{\partial f_i}{\partial c}$$

$$f_{di} = \frac{\partial f_i}{\partial d}$$

$$f_{ei} = \frac{\partial f_i}{\partial e}$$

$$f_i = f(x_i, y_i, a', b', c', d', e')$$

Substituting Eq. (A.6) into Eq. (A.4) produces a set of linear equations which can be easily solved for Δa, Δb, Δc, Δd, and Δe.

Using these equations, an iterative method may be used to solve for the parameters of the ellipse. The algorithm starts by calculating the deltas using Eq. (A.6) and (A.4), and an initial guess of the parameters of the ellipse. Then, the guess is updated by the deltas to obtain better parameter estimates and the deltas are recalculated using Eq. (A.6) and (A.4) and the new parameters. The process is repeated until the deviation is small enough to indicate that a', b', c', d' and e' are acceptable estimates of the true parameters of the ellipse.

5. References

1. A.J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of Human Faces," *Proceedings of the IEEE,* vol. 59, No. 5, May 1971, pp. 748-760.

2. L.D. Harmon and W.F. Hunt, "Automatic Recognition of Human Face Profiles," *Computer Graphics and Image Processing,* 6, 1977, pp. 135-136.

3. L.D. Harmon, S.C. Kuo, P.F. Famig, and U. Raudkivi, "Identification of Human Face Profiles by Computer," *Pattern Recognition,* vol. 10, 1978, pp. 301-312.

4. L.D. Harmon, M.K. Khan, R. Lasch, and P.F. Ramig, "Machine Identification of Human Faces," *Pattern Recognition,* vol. 13, No. 2, 1981, pp. 97-110.

5. T. Sakai, M. Nagao and Takeo Kanade, "Computer Analysis and Classification of Photographs of Human Faces," *Proceedings of the First USA-Japan Computer Conference,* 1972, pp. 55-62.

6. L.K. Bromley, "Computer-aided Processing Techniques for Usage In Real-time Image Evaluation," *Master's Thesis,* University of Houston, 1977.

7. M.L. Gillenson and B. Chandrasekaran, "A Heuristic Strategy for Developing Human Facial Images on a CRT," *Pattern Recognition,* vol. 7, 1975, pp. 187-196.

8. T.P. Wiederhold, "Facial Image Generation By Computer," *Master's Thesis,* University of Houston, 1976.

9. A.M. Alattar and S.A. Rajala, "Primitive-Based Teleconference Image Coding Technique," *ICASSP 1989,* Accepted for publication.

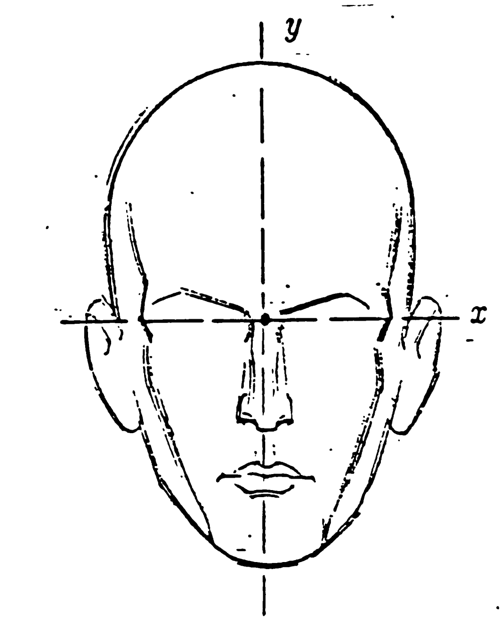
10. B. Hogarth, *Drawing the Human Head, 1st ed.,* Watson-Guptill, New York 1965.

11. S. Simmons III and M.S. Winer, *The Creative Process,* Prentice-Hall, New Jersey 1977.

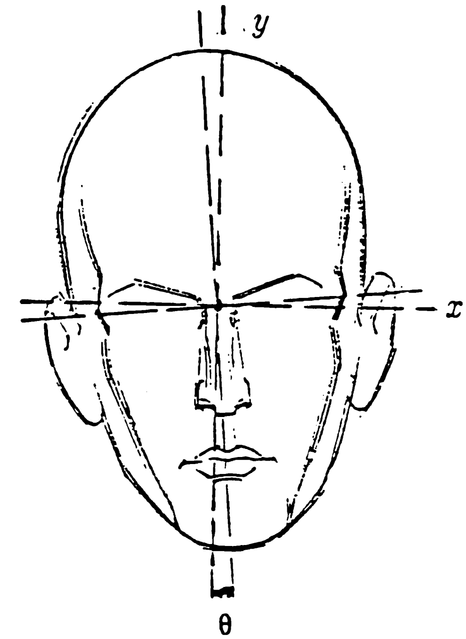
12. D.H. Ballard and C.M. Brown, *Computer Vision,* Prentice-Hall, New Jersey, 1982.

13. E.L. Hall, *Computer Image Processing and Recognition,* Academic Press, New York, 1979.

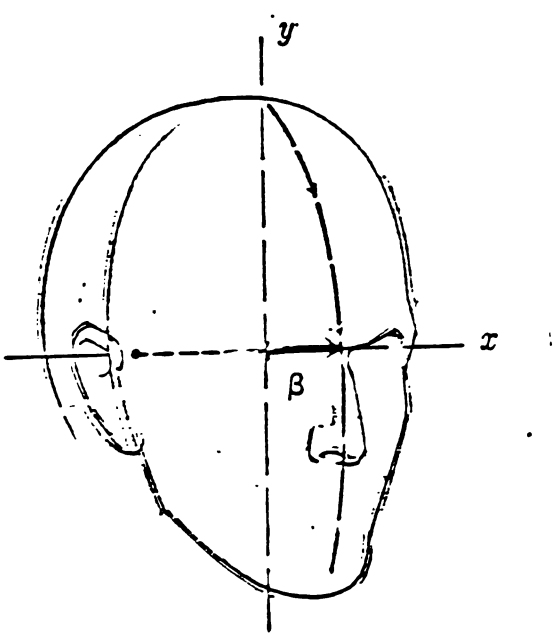
14. R.T. Chen, H.K. Wan, D.L. Stover, and R.D. Iverson, "A One-Pass Thinning Algorithm and Its Parallel Implementation," *Computer Vision, Graphics, and Image Processing,* vol. 40, Oct. 1987, pp. 30-40.



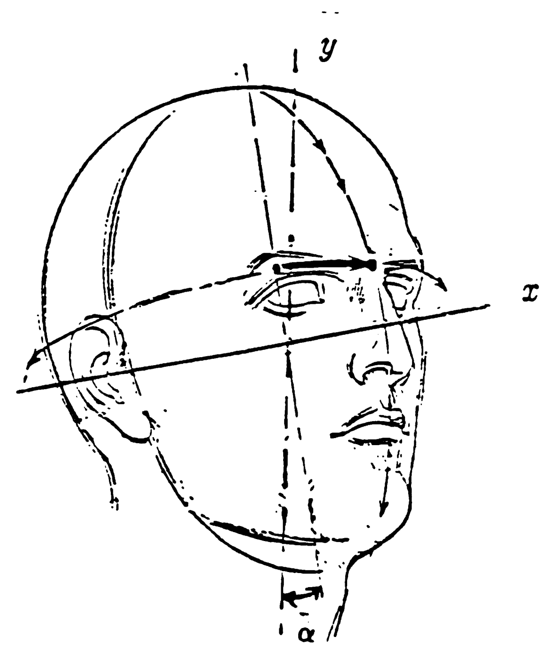
a. Normal Position



b. Rotation Around Z-axis



c. Rotation Around Y-axis



d. Rotation Around X-axis

Fig. 1. Head Rotation Angles With Respect to *x, y, and z* axes

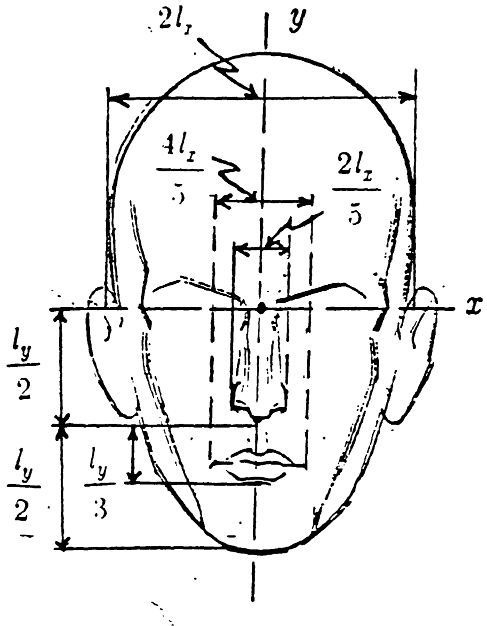


Fig. 2. Head Model For Head and Shoulder Images

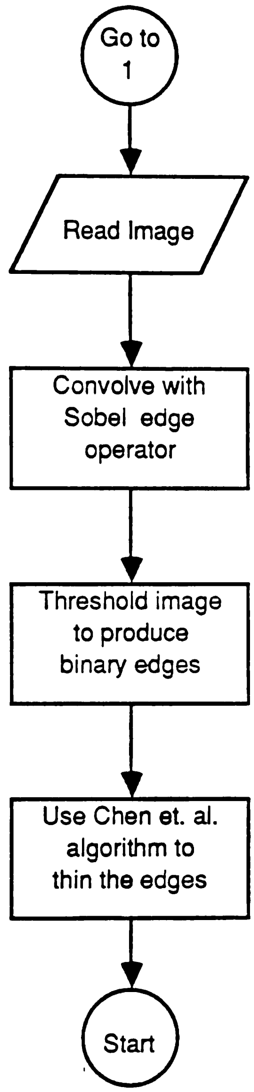


Fig. 3a. Flowchart for Edge Detection

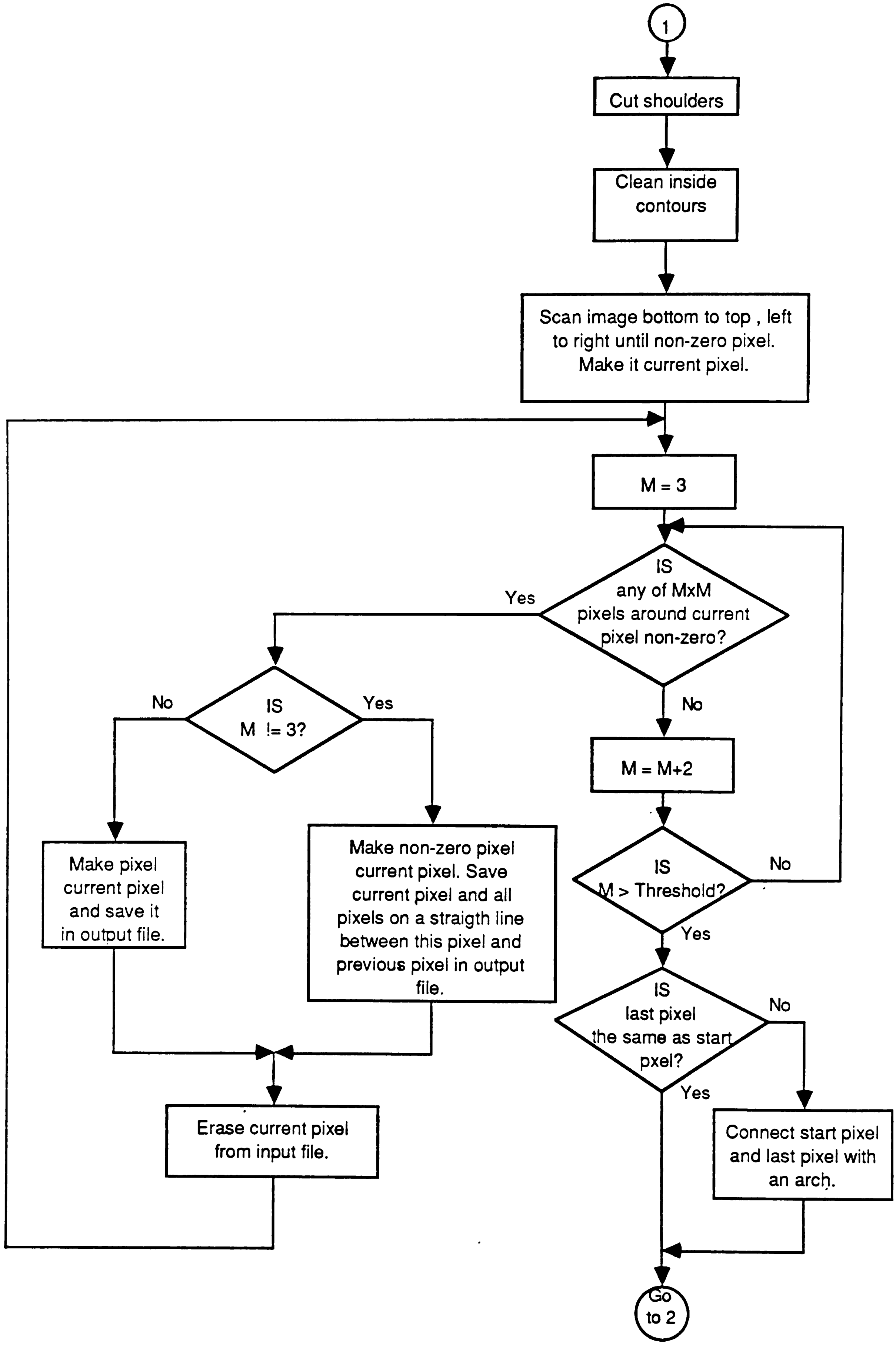


Fig. 3b. Flowchart for Head Contour Extraction

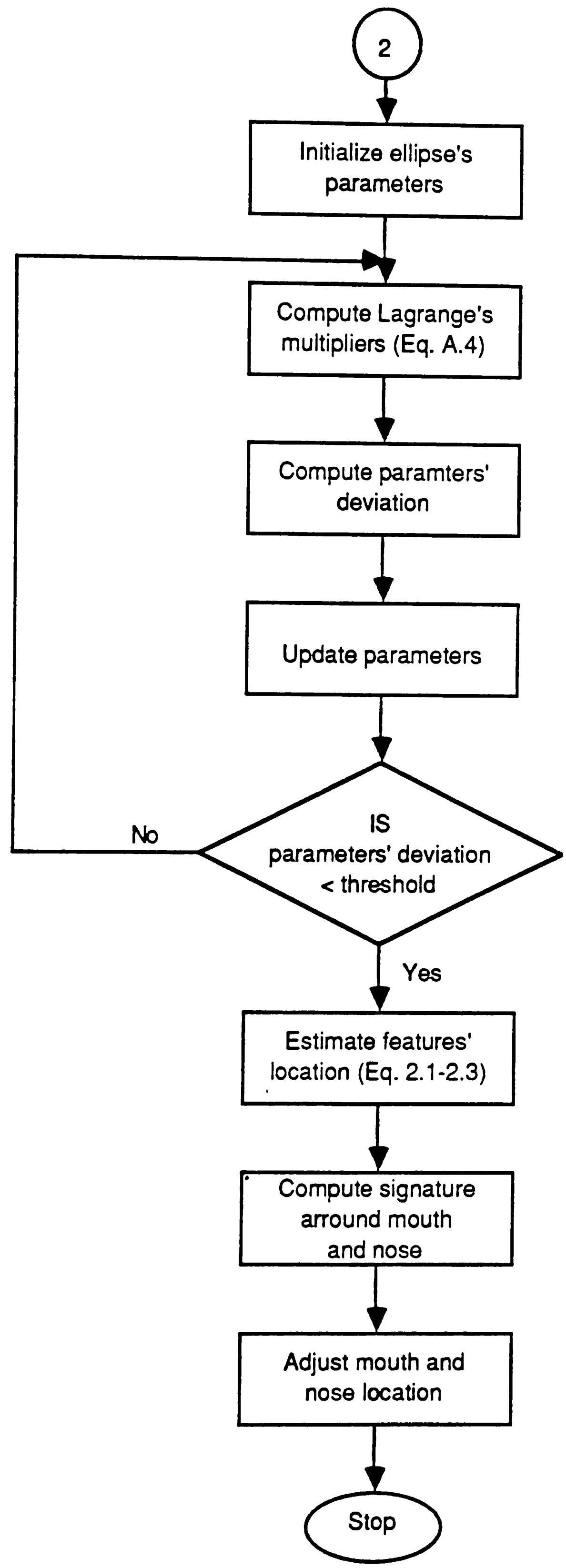


Fig. 3c. Flowchart for Head Feature Location

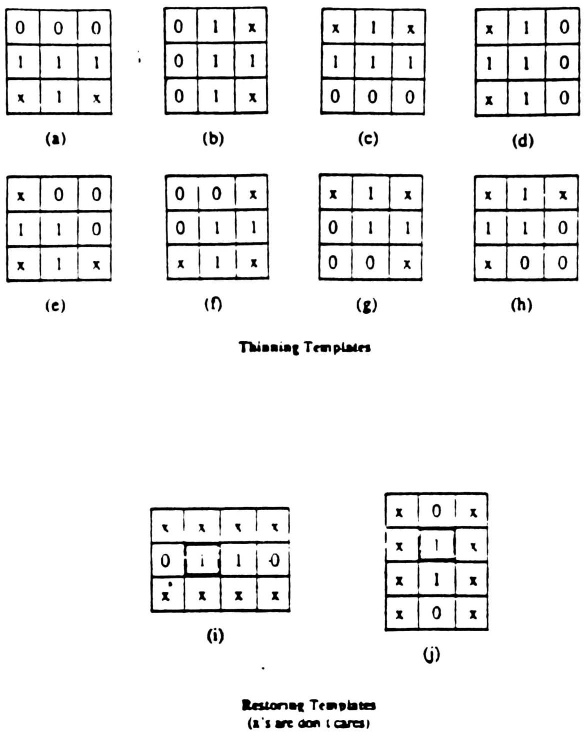


Fig. 4. Chen et. al. Templates

Fig. 5. 128×128 Input Image

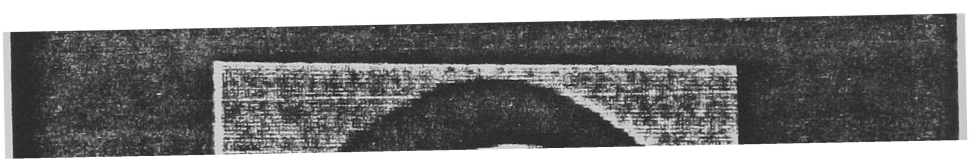


Fig. 6a. The Edge of the Image in Figure (5)

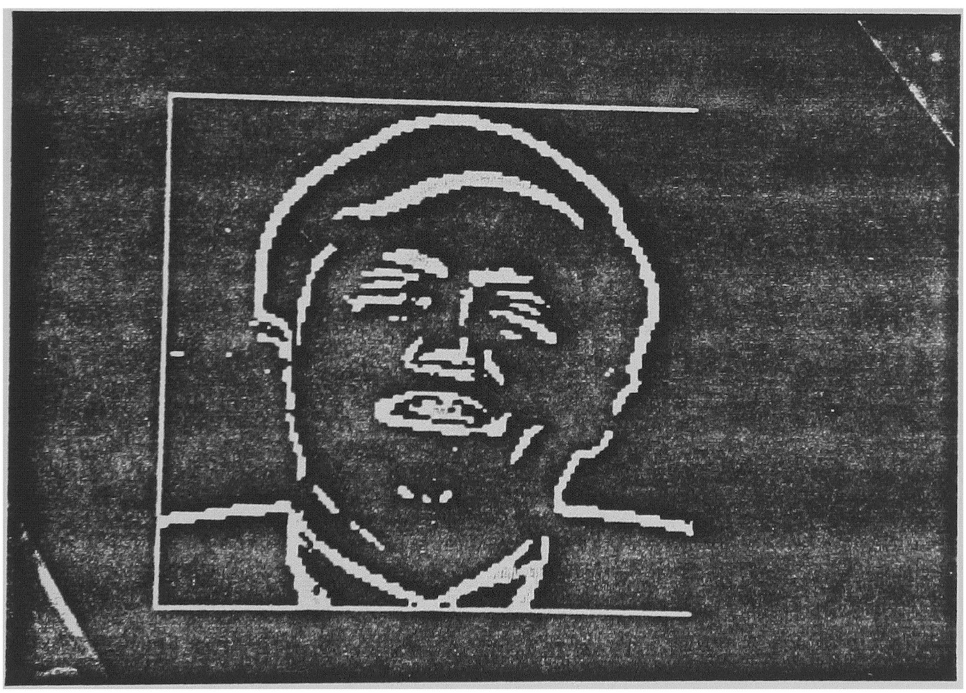


Fig. 6b. The Result of Thresholding the Image in Figure (6a)

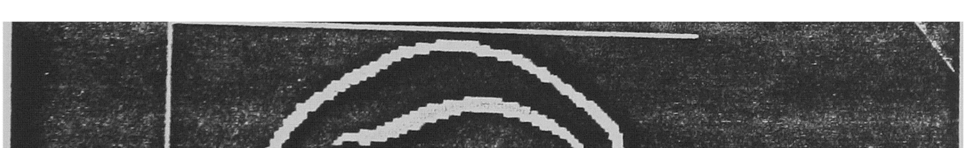


Fig. 6c. The Result of Thinning the Image in Figure (6b)

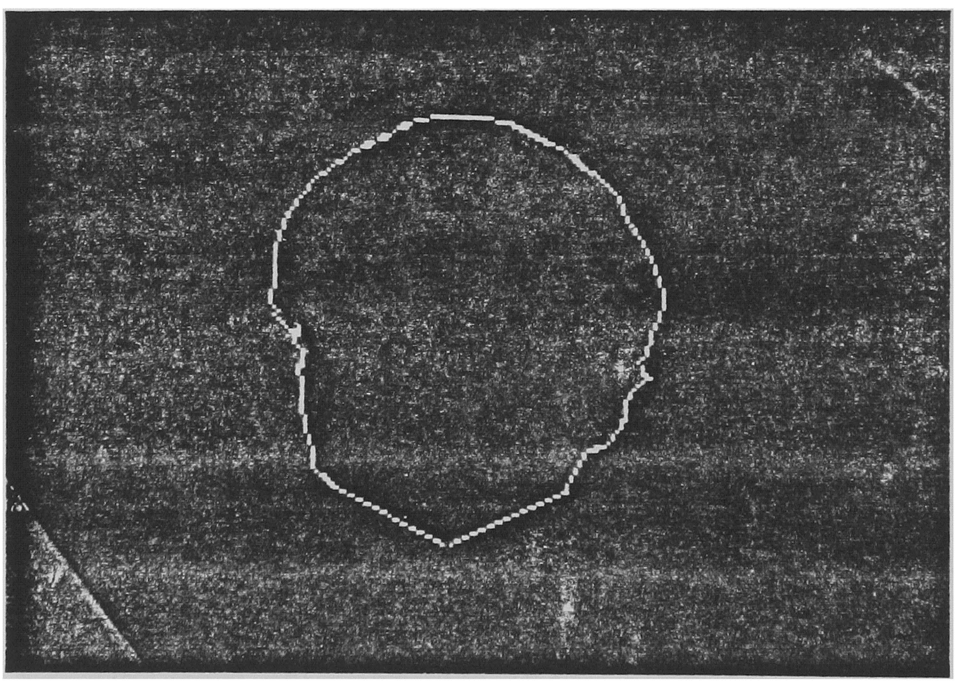
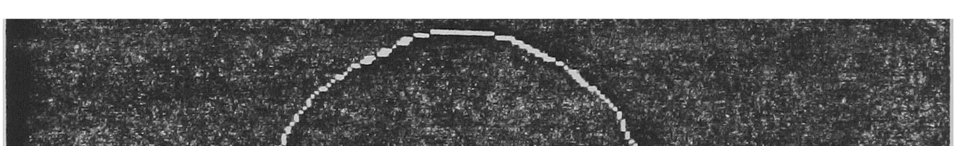


Fig. 6d. Outer Most Head Contour of the Image in Figure (6c)



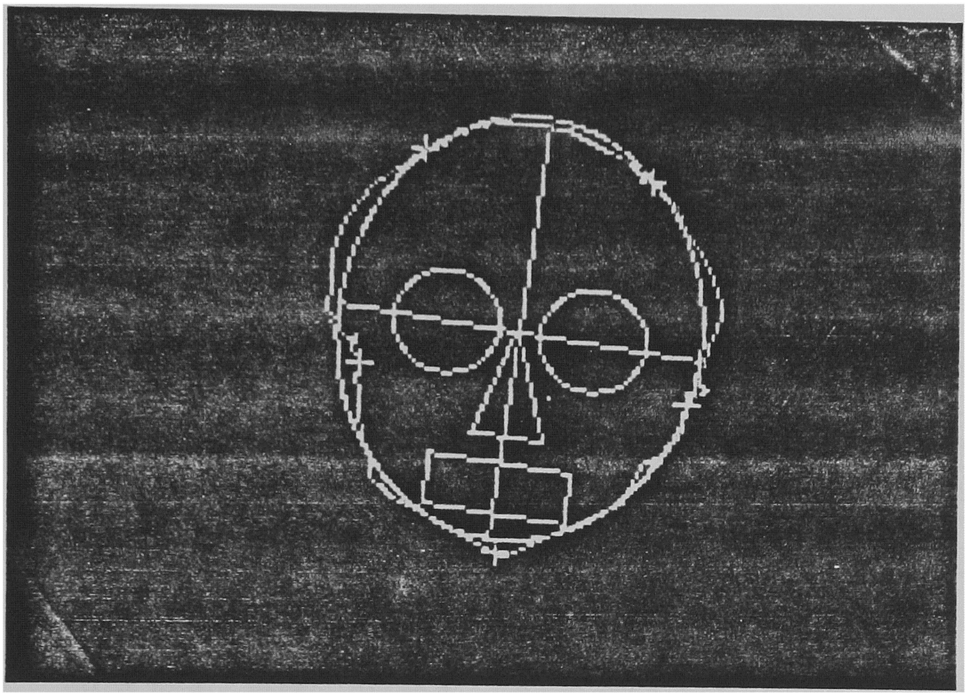
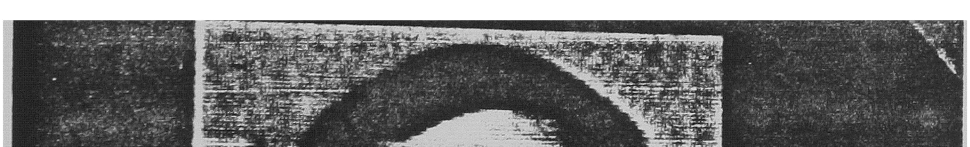


Fig. 6e. Estimated Ellipse Which Fits the Head Contour of Figure (6c)

Fig. 7. Locations of the Four Major Facial Features in Figure (5)



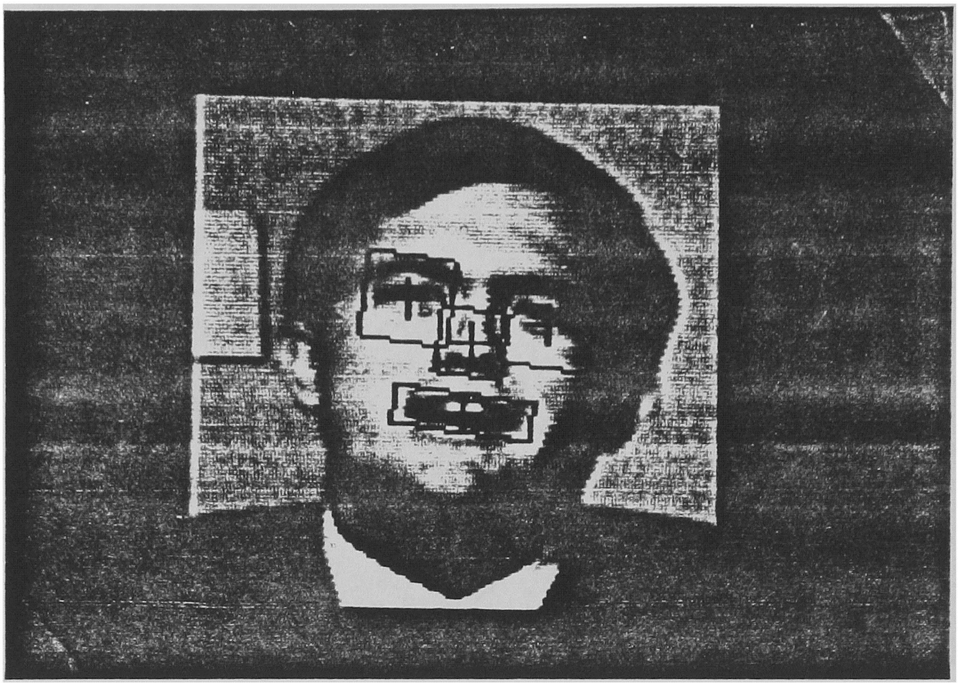


Fig. 8. Adjusted Location of the Four Major Facial Features in Figure (5)