Experiments in Detecting Face Contours *

XIAOBO LI
li@cs.ualberta.ca

NICHOLAS ROEDER
roeder@cs.ualberta.ca

Department of Computing Science
University of Alberta, Edmonton
Alberta, Canada T6G 2H1

Abstract

This paper reports on experiments in detecting face contours from front-view ID-type pictures. Based on the eye and mouth positions, which can be detected by the algorithms proposed in [5], a simplified Adaptive Hough Transform (AHT) technique is used to identify approximately vertical straight lines of the cheeks from the edge image. Independently, parabolas forming the chin are detected by another AHT procedure. The location and curvature of the chin line, together with the cheek lines, characterize the shape of the face. This method was tested on over 70 different face images and is shown to produce results with a high degree of accuracy. We discuss in detail the specific issues involved in the detections, such as the definition of relevant subimage, parameter ranges, resolution of the accumulator array, peak cells, and end point determination.

1 Introduction

There are basically two approaches to computer recognition of human faces: face-based and constituent-based. The face-based approach treats a face as a 2-D pattern of intensity variation, and attempts to capture and define the face as a whole. The constituent-based approach examines individual structural constituents of the face and takes numerical measurements of them [6, 4, 7, 13, 9, 14].

In the studies referenced above, the deformable template technique has been heavily used. With this method, locating a feature is accomplished through standard function optimization techniques. Since numerical optimization cannot guarantee a global optimum, it generally yields a *good* solution but not necessarily the *best* one. In searching for the best solution, one might turn to a more exhaustive method, such as Hough Transform (HT), also known as parameter transform [3] or parameter hashing [8].

Several attempts have been made to reduce the computation load and the storage requirement of the Hough transform [11]. The Adaptive Hough Transform (AHT) scheme was proposed by Illingworth and Kittler [10], and is shown to be 5 times faster than the standard implementation and uses an accumulator which is 3000 times smaller in normal cases.

We divide the face contour into cheeks and chin, and use a simplified AHT to detect straight lines for the cheeks, and parabolas for the chin. Our experiments are discussed in detail in the remainder of this paper.

2 Simplified AHT

The points (usually edge points) on a curve characterized by n parameters, $a_1, \ldots, a_n$, can be defined by an equation of the form

$$f((a_1, \ldots, a_n), (x, y)) = 0.$$

When straight lines are considered, $n = 2$. In cheek line detection, we use r and A as the two parameters. In chin line detection, three parameters c, x_0 and y_0 are used to describe parabolas. The Hough Transform examines all edge pixels in a *relevant subimage* W of the input image. Any edge pixels outside of W are irrelevant to the curve detection process. The standard implementation of HT involves representing the continuous parameter space by an appropriately quantized space which is often referred to as an *accumulator array*. The edge points in the image can provide evidence for, or vote for, a model instance which has parameter values falling in a particular cell of the accumulator array.

*This research is supported in part by the Canadian Natural Sciences and Engineering Research Council under Grant OGP9198 and a SEED grant from the federal government of Canada.

The traditional HT technique usually requires a significant amount of computation time and storage space. Adaptive Hough Transform, which greatly increases the computation speed and reduces the storage requirement, is an iterative procedure which refines the search area in the parameter space until the target precision is achieved. In iteration i, the search area is

$$S^{(i)} = L_1^{(i)} \times L_2^{(i)} \times \cdots \times L_n^{(i)}$$

where $L_j^{(i)}$ is an interval (U_j, V_j) of possible values of the jth parameter a_j. In general, for all j, $L_j^{(i+1)}$ is smaller than $L_j^{(i)}$, and for a certain i, the search area $S^{(i)}$ will reach the target precision for some or all parameter dimensions. In our study, all parameters reach the target precision at approximately the same iteration, and then a particular parametric curve in the image space can be identified.

Before applying the AHT, a simple 3×3 Sobel edge detection routine generates an edge map from the input face image. A thresholding process is applied to the non-zero edge strengths.

The simplified AHT procedure is as follows.

```
Let i = 0;
Set S^(i) to be the set of all possible
parameter combinations;
While target precision not reached {
    Divide S^(i) into m_1 × ··· × m_n cells;
    For each edge pixel {
        Accumulate the corresponding
        parameter cells;
    }
    Find the peak in the accumulator;
    i = i + 1;
    Set S^(i) to be the peak and its
    neighborhood;
}
```

This procedure detects a single parametric curve in the image.

Every particular application of AHT must deal with certain specific issues. We deal with the following issues, some of which are common to other applications of AHT.

Relevant subimage (W) A portion of the input face image is identified in order to detect the desired features and only the edge pixels within it will be used in the remaining process. In our face contour detection algorithms, the positions of the eyes and the mouth are used to determine W.

Parameter ranges The search area of the parameter space in the initial iteration $S^{(0)}$ needs to be determined for each particular application.

Resolution of the accumulator The resolution of the accumulator array, which greatly affects the computation speed, must be decided upon. In [10], $m_1 = m_2 = 9$.

Edge strength and angle information This information may be used in the accumulation process. The angle information has been used previously in some applications [1] to limit the number of parameter cells to increment. The edge strength E is sometimes used in the accumulation operation of the vote Z of a parameter cell, i.e.

$$Z := Z + E + C, \tag{1}$$

where C is a constant. A small C value will emphasize the weighting of the edge strength. In our experiments, it was found that setting C to 0 produced the most acceptable results because the effects of noise edges could be reduced.

Neighborhood of a peak A neighborhood of the detected peak cell must be chosen in order to capture the "true" peak. The size and shape of the search neighborhood should be carefully selected to balance the performance and computation speed. The AHT method allows the size and shape of the neighborhood to change at each iteration, but we modify the AHT to use a fixed size. Since the accuracy of detection is affected mainly by the quality of the edge image, our simplification is justified. Using the original AHT may require more time.

End points of the result curve When the target precision is reached, a single curve in the image is detected, which would be the end of processing in some applications. In our face contour extraction, we also need to know the ending points of the curve in the image.

Each of these issues involves trade-offs of certain kinds. For example, a smaller search area in the parameter space allows it to be more finely divided and can save computation time, but a larger range will cover more cases, increasing reliability. The choice of accumulator resolution and the size of peak neighborhood affect the the number of iterations k required to reach the target precision q. Consider a one-dimensional case. For the parameters x_0 and y_0 in the parabola, q is 1 pixel. Let m denote the number of cells the accumulator is divided into, L denote the size of the search area, and b denote the neighborhood size. The number of required iterations k will be

$$k = \frac{\log(qm/L)}{\log(b/m)}. \tag{2}$$

One purpose of our experiments was to identify these issues and decide on good compromises. The next two sections report some of the decisions made on these issues in the cases of cheek and chin detection.

3 Cheek Detection

This section outlines the procedure for detecting cheek lines using the eye and mouth locations provided by the algorithm in [5]. A simplified AHT algorithm was employed to detect approximately vertical, straight lines.

3.1 The Equation and Parameters

In a front-view image of a face, the cheek lines are approximately straight, and almost always close to vertical in orientation. We use the standard polar coordinate equation

$$r = x \cos A + y \sin A. \quad (3)$$

The parameter space is two dimensional, and the value of r is calculated for each A value to determine the accumulator cell for each edge pixel. The value corresponding to the middle of the cell is considered to be the value for that cell.

We now report our findings in the decision making on the issues mentioned in Section 2.

Relevant subimage The methods in [5] can provide the positions of the centers of the two eye irises and the vertical position of the mouth, as shown in Figure 1. Two rectangular regions of the input image are used. The vertical search area covers from the pupil to the mouth, and the horizontal range is between $l_1 w$ and $l_2 w$ units away from the left iris. The search area for the right cheek is defined similarly. A conservative estimate was used initially to set l_1 through l_4 and the true values of the l's were recorded in each test case in order to facilitate the parameter setting in future studies. Table 3.1 reports some statistics of the l values from the 81 test images.

Values	l_1	l_2	l_3	l_4
extreme	66.7%	40.0%	38.9%	69.1%
mean	57.6%	55.1%	56.3%	57.9%

Table 1: Recorded parameter values for the cheek regions

Parameter ranges Since the detection of close to vertical lines is necessary, the range of A to be searched is from 80 degrees to 100 degrees. The range of r is set according to the relevant subimage. The fact that the range for A is so close to vertical allows the assumption that the minimum r value corresponds to a line going through the lower left hand corner of the relevant subimage with A as 100 degrees. Also, the maximum r value corresponds to a line going through the lower right corner with an angle of 80 degrees, i.e.,

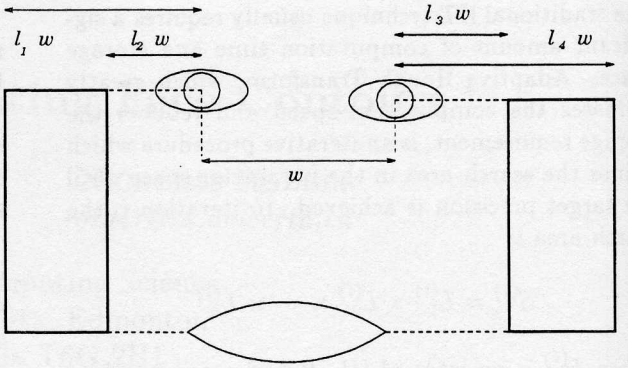

Figure 1: Relevant subimages for cheek detection

$$r_{max} = (x_{max} - y_{max} \text{ctan} 80^o) \cos 80^o + \frac{y_{max}}{\sin(80^o)},$$

$$r_{min} = (y_{min} - x_{max} \text{tan} 10^o) \cos 10^o.$$

Resolution of the accumulator The size of the accumulator was chosen to be 8×8. In this application, a larger accumulator will cause each iteration to require more computation. Using a smaller accumulator causes the number of iterations to grow significantly because of the large range of each cell. Our experiments showed that the required precision of both parameters could often be reached in the same number of iterations when using a square accumulator.

Edge strength and angle information The angle of each edge pixel is used to decide on the A value of the corresponding line. The form of the line equation allows direct use of the edge angles. Recall that with the definition of angle A, the edge pixels with angle α can only contribute to the straight lines of the form $r = x \cos \alpha + y \sin \alpha$. In practice, one more cell on either side of the calculated angle cell is also accumulated. This way, only a portion of the accumulator is covered, reducing computation while maintaining reasonable accuracy.

Neighborhood of a peak In our experiments, a 3×3 neighborhood surrounding the peak cell is used.

End points of the result curve Among the

edge pixels associated with the detected line, the ones with the extreme x (or y) values would normally be considered as the end points. To find the endpoints, we use the following simple technique:

```
for each x_i in subimage {
    calculate y_i range based on A range;
}
set endpoint_1 = second edge within range;
set endpoint_2 = second last edge within
range;
```

3.2 Results and Discussion

The cheek detection procedure has been tested using 81 face images. Some faces are relatively large compared to the whole image, while some are quite small, as shown in Figure 2 . The individuals are of different races and skin tones. Some have a beard or glasses.

Figure 2 gives examples of the cases where the cheek detection was successful on both sides of the face. Examples of "marginal" and "wrong" fits are shown in Figure 3. The presence of a beard and glasses in Figure 2 did not affect the detection.

Table 3.2 shows the test results. Note that the cheek detection procedure located both cheeks with at least a marginal fit in 63 images. If a score of 2 is given to a "good" fit, 1 to a "marginal" fit, and 0 to a "wrong" fit, our score is 270, which gives 83.3% of the maximum possible scoring.

Case	No.	Score
(good,good) Both cheek lines are correctly located	51	204
(good,marginal) One "good"-fit and one "marginal"-fit	10	30
(marginal,marginal) Approximate locations are detected on both sides	2	4
(good,wrong) One cheek position is correct, the other is wrong	14	28
(marginal,wrong) One cheek position is approximate, the other is wrong	4	4

Table 2: Cheek line detection results

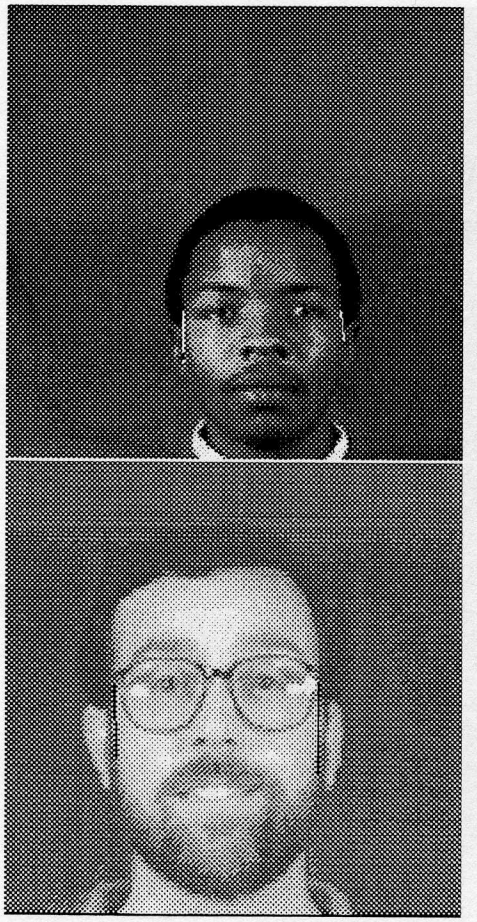

Figure 2: Examples of "good" cheek detection.

The experiments were run on a Sparc IPC with a benchmark performance of 13.4 SPECmarks89. The average total time for cheek detection is 3.0 seconds (after edge detection). About 2.1 seconds of this time were spent reading/writing files.

4 Chin Detection

Detection of chin lines involves searching for parabolas in the edge image.

4.1 The Equation and Parameters

Normally, a parabola is characterized by an equation with four or more parameters [5]. Choosing an equation with the lowest dimensionality reduces the algorithmic complexity of the HT exponentially. In a front-view face image, most chins can be represented by an upright parabola, i.e., the central axis of the parabola is vertical. Thus, the form in Figure 4 is chosen.

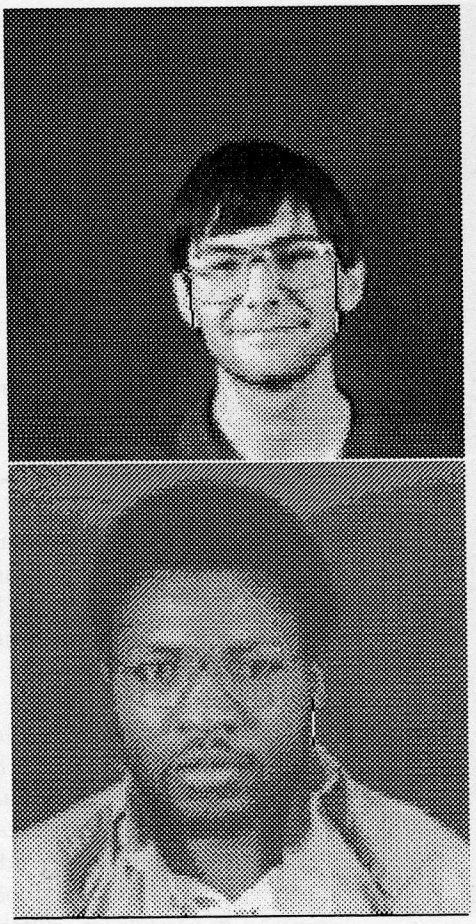

Figure 3: Examples of mixed cheek results

We now report our decisions on the issues mentioned in Section 2 in the case of chin detection.

Relevant subimage A rectangular subimage under the mouth is defined based on the position of the eyes and mouth. The position and the shape of the rectangle is determined by four variables, l_1 through l_4, defined in Figure 5. The actual l values (as a percentage of w), observed from 28 test images, are recorded in Table 4.1. From our experiments on the remaining images, simply setting both l_1 and l_2 to zero produces long enough chin lines, greatly simplifies the end point determination, and also reduces the computational load.

Parameter ranges The ranges of x_0 and y_0 are defined according to the rectangle in Figure 5. The initial range of c was set from -0.09 to -0.002. It has been observed that the minimum c value is -0.00223 and the maximum is -0.02973, which suggests that the range for c could be restricted further. The target precision of the c value is set to 0.0005, while the target precision for x_0 and y_0 is a single pixel.

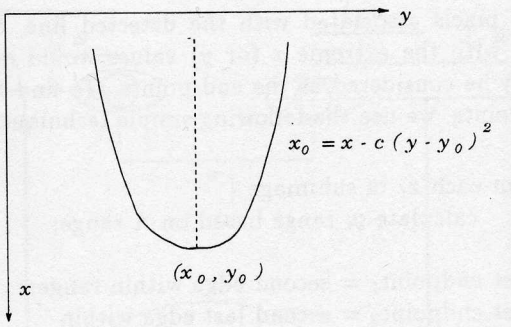

Figure 4: Parameter definitions in parabola detection

	l_1	l_2	l_3	l_4
max	77.1	66.7		97.6 (no beard), 111.3 (w/beard)
min			20.0	

Table 3: Recorded parameter values for the chin regions (as percentages)

Resolution of the accumulator Recall Equation 2. The search area for c is $L = 0.088$. Assuming $b = 3$ and choosing $m = 6$ will require $k = 5$ iterations to reach the target precision $q = 0.0005$. The value 6 for m_x and m_y also gives approximately 5 iterations. Therefore, the search area $S^{(i)}$ is divided into $6 \times 6 \times 6$ cells for all i.

Neighborhood of a peak A $3 \times 3 \times 3$ neighborhood surrounding the peak cell is used for zooming in.

End points of the result curve When the parabola is being drawn on the image, the endpoints are found by looking from the extreme horizontal limits of the image space and moving towards the calculated vertex of the parabola until a group of at least 8 connected pixels (in the edge image) are found. Then, the extreme point of this group of 8 is

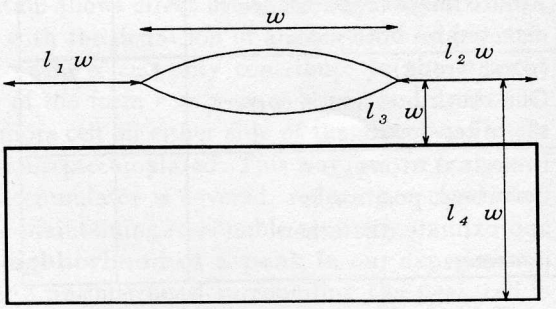

Figure 5: Relevant subimages for chin detection

used as the end point. The number 8 was obtained from experiment.

4.2 Results and Discussion

The chin detection procedure has been tested using 72 face images in which some edge points are present around the chin area. The results are categorized as "good", "marginal" and "wrong" in a similar way to that of cheek detection. Table 4.2 shows the test results. Note that the chin detection procedure successfully located the chin with at least a marginal fit in 67 images (93.1 percent of the cases).

Case	No.	Description
good	58	The chin line is located correctly
marginal	9	The parabola(s) fit(s) the chin approximately
wrong	5	The parabola(s) is (are) wrong

Table 4: Chin line detection results

Figure 6 gives examples of good fit. Some of the "marginal" results are shown in Figure 7. Figure 8 shows examples of "wrong" results in chin detection.

The average total time for chin detection is 6.0 seconds for finding one parabola line. If there is another parabola to be detected, the processing time is about 10.0 seconds.

The major limitation of this chin detection procedure is that it heavily relies on the result of edge detection, as do many other Hough-based methods. In Figure 9 the top image shows an example in which the edge information is sufficient, whereas the bottom image contains inadequate information.

Figure 10 gives a general impression of cheeks and chin together. A boundary following procedure, such as the one in [4] could be used to connect the cheeks and chin together to form a complete face contour. However, the cheek and the chin lines do not need to be linked for recognition. The position of the cheek lines give enough information to calculate the width and length of the face, and the angle of cheeks and the curvature of the chin already characterize the shape and orientation of the face.

5 Conclusion

Automatic detection of facial features is crucial in face recognition. It is also a challenging and interesting pattern recognition problem which has led

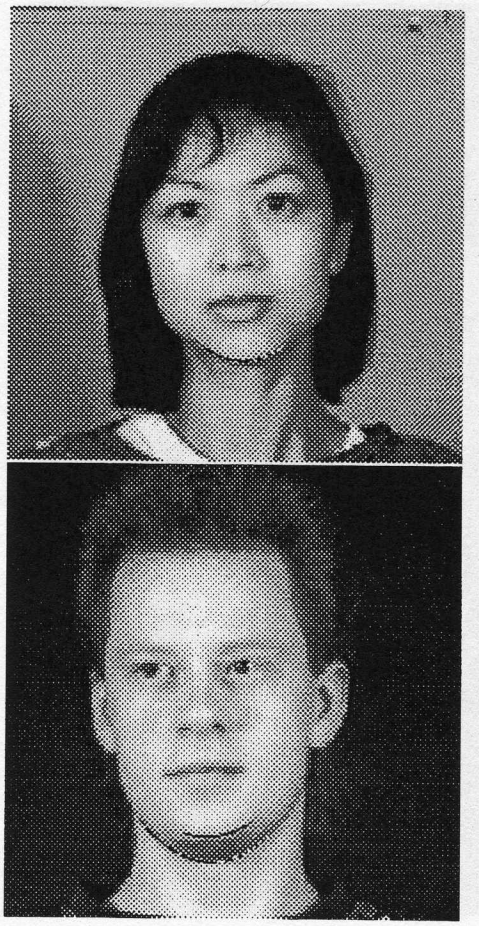

Figure 6: Examples of "good" results from chin detection

to many important theoretical and practical developments useful in more general applications. Since the well known deformable template technique may not generate a global optimum, this experiment attempted to use a more exhaustive method, the Hough Transform. A simplified version of the AHT proposed in [10] is employed and many specific issues involved in the implementation are discussed in detail.

The major limitation of this technique is the dependence on the quality of edge detection. Low lighting conditions and low image quality severely reduce the edge quality, and hence damage the face contour detection. Other techniques are needed, such as using the direction of the maximum intensity change to detect noses, with no clear edge present [12]. One of our ongoing projects is to detect face contours using information other than edges. A combination of iso-intensity map [2] and gradient direction could be used as input to the AHT procedure, which could be kept relatively unchanged.

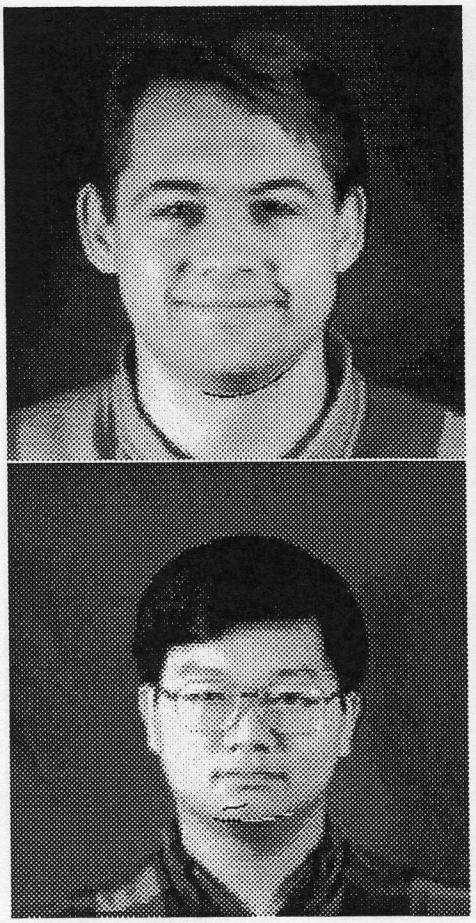

Figure 7: Examples of "marginal" result of chin detection

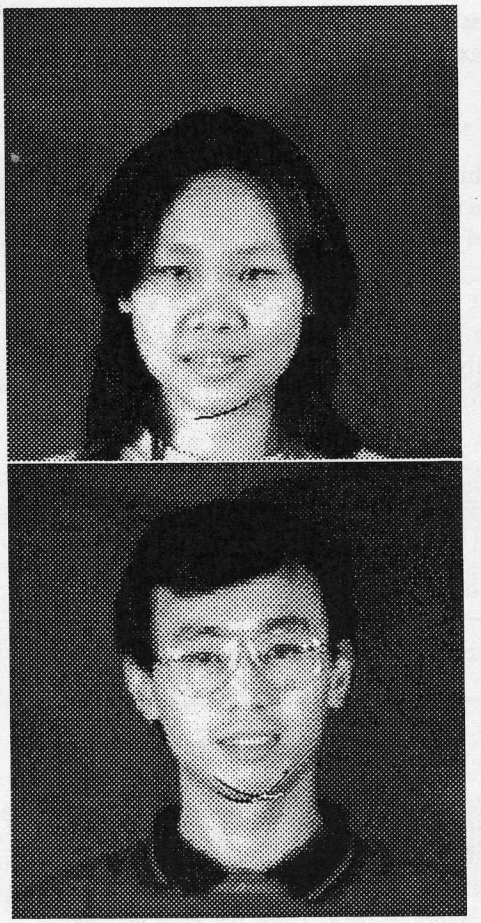

Figure 8: Examples of "wrong" results from chin detection

Other avenues for further research should closely relate to the recognition process in order to determine the precision requirement of the AHT, which would be another step towards a complete model-based system [5].

References

[1] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] Terry Caelli. Private communication.

[3] A. Califano and R. Bolle. The multiple window parameter transform. *IEEE Trans. on Patt. Anal. and Machine Intell.*, 14(12):1157–1170, 1992.

[4] G. Chow and X. Li. Experiments in detecting facial features. *Proc. Vision Interface '93*, pages 157–164, 1993.

[5] G. Chow and X. Li. Towards a system for automatic facial feature detection. *Pattern Recognition*, to appear.

[6] I. Craw, H. Ellis, and J.R. Lishman. Automatic extraction of face-features. *Pattern Recognition Letters 5*, pages 183–87, 1987.

[7] V. Govindaraju, S.N. Srihari, and D.B. Sher. A computational model for face location. *Proceedings of the 3rd Int'l Conference on Computer Vision*, pages 718–721, 1990.

[8] W. Grimson and D. Huttenlocher. On the sensitivity of the Hough transform for object recognition. *IEEE Trans. on Patt. Anal. and Machine Intell.*, 12(3):255–274, 1990.

[9] C-L. Huang and C.-W. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition*, 25(12):1435–1444, 1992.

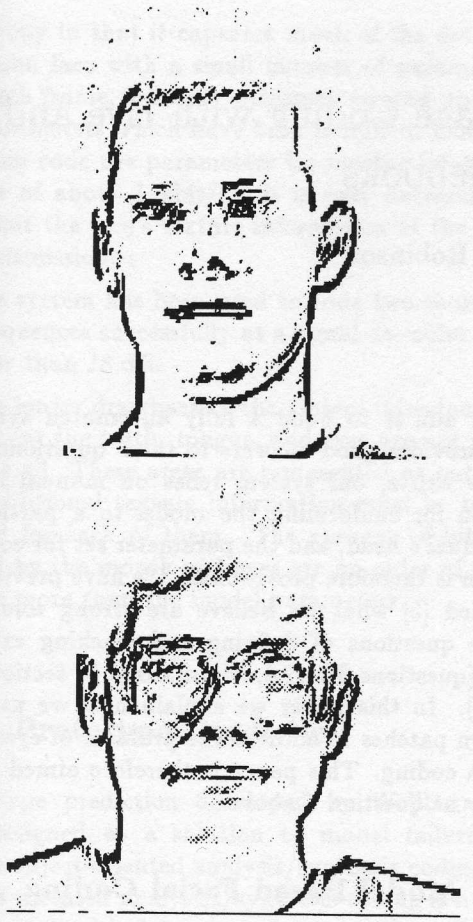

Figure 9: Images with and without sufficient edge information for chin detection

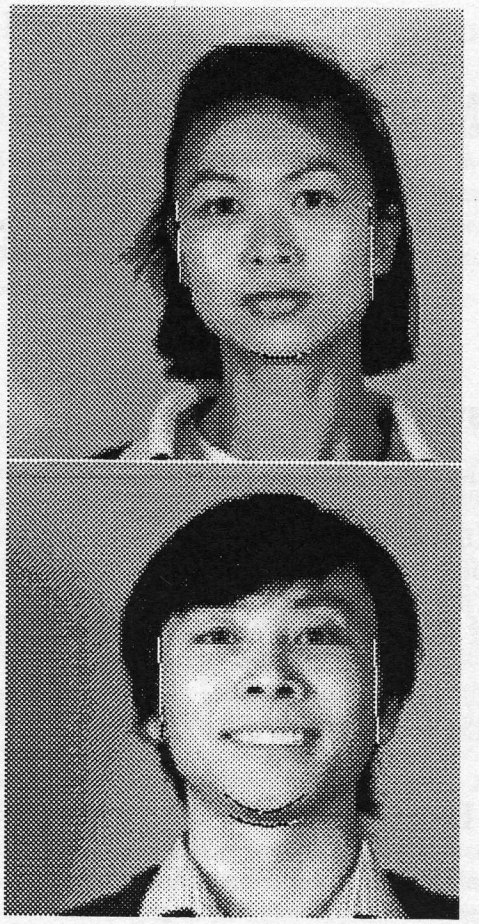

Figure 10: Cheek and chin detection results together

[10] J. Illingworth and J. Kittler. The adaptive Hough transform. *IEEE Trans. on Patt. Anal. and Machine Intell.*, PAMI-9(5):690–698, 1987.

[11] H. Li and M.A. Lavin. Fast Hough transform based on bintree data structure. *Proc. Conf. Computer Vision and Pattern Recognition*, pages 640–642, 1986.

[12] Graham Robertson and Ken Sharman. Object location using proportions of the direction of intensity gradient (prodigy). *Proceedings of Vision Interface '92*, pages 189–194, May 1992.

[13] Toshiyuki Sakai, Makoto Nagao, and Takeo Kanade. Computer analysis and classification of phototgraphs of human faces. *First USA-Japan Computer Conference Proceedings*, pages 55–62, October 1972.

[14] Alan Yuille, David Cohen, and Peter Hallinan. Facial feature extraction by deformable templates. Technical Report 88-2, Harvard Robotics Laboratory, 1988.