International Journal of Computer Vision 19(2), 129–146 (1996)

Locating Human Faces in Photographs

VENU GOVINDARAJU*
Center of Excellence for Document Analysis and Recognition (CEDAR), Department of Computer Science, 226 Bell Hall, State University of New York at Buffalo, Buffalo, NY 14260
govind@cedar.buffalo.edu

Abstract. The human face is an object that is easily located in complex scenes by infants and adults alike. Yet the development of an automated system to perform this task is extremely challenging. An attempt to solve this problem raises two important issues in object location. First, natural objects such as human faces tend to have boundaries which are not exactly described by analytical functions. Second, the object of interest (face) could occur in a scene in various sizes, thus requiring the use of scale independent techniques which can detect instances of the object at all scales.

Although, the task of identifying a well-framed face (as one of a set of labeled faces) has been well researched, the task of locating a face in a natural scene is relatively unexplored. We present a computational theory for locating human faces in scenes with certain constraints. The theory will be validated by experiments confined to instances where people's faces are the primary subject of the scene, occlusion is minimal, and the faces contrast well against the background.

1. Introduction

The human face is an object that is easily located in complex scenes by infants and adults alike. Our objective is to locate faces in photographs such as the ones typified by those found in newspapers. By locating a face is meant the framing of the face by means of a rectangle such that the frame includes as much of the face as possible and includes as little of other material as possible. Figure 1 illustrates the behavior of a desired face locating system. Figure 1(a) shows a cluttered image which is a typical input to the system. Figure 1(b) is the output of the face locator.

Simplifying assumptions must be made to make the problem tractable. The range of variability in poses[1] and sizes of faces should be confined to a small range. A reasonable constraint is to consider only those views which capture at least one of the eyes (near-frontal view). A large percentage of front-page newspaper photographs naturally adhere to these constraints. People referred to by the caption invariably provide a near-frontal view as readers are expected to identify them. Even the sizes of the faces must lie within a certain range so as to conform to the aesthetic sense of readers. Furthermore, the caption provides valuable knowledge about the number of people featuring in the photograph and the spatial relations between them. On the other hand, several objects with shapes similar to the face are often cluttered in a scene. Uncertainty can arise from the extensive variability in the number of objects in the scene, their orientations and relative positions, and noise related to poor paper quality and digitization.

Previous systems have targeted face specific features like eyes from the very beginning and therefore fail in cases when the background is cluttered and when there are many people in the photograph. Our approach is to divide the problem into two stages. In the first stage, candidate regions are hypothesized by "looking" for gross features. In the second stage the candidate

*This work was supported in part by the NSF and The Eastman Kodak Company.

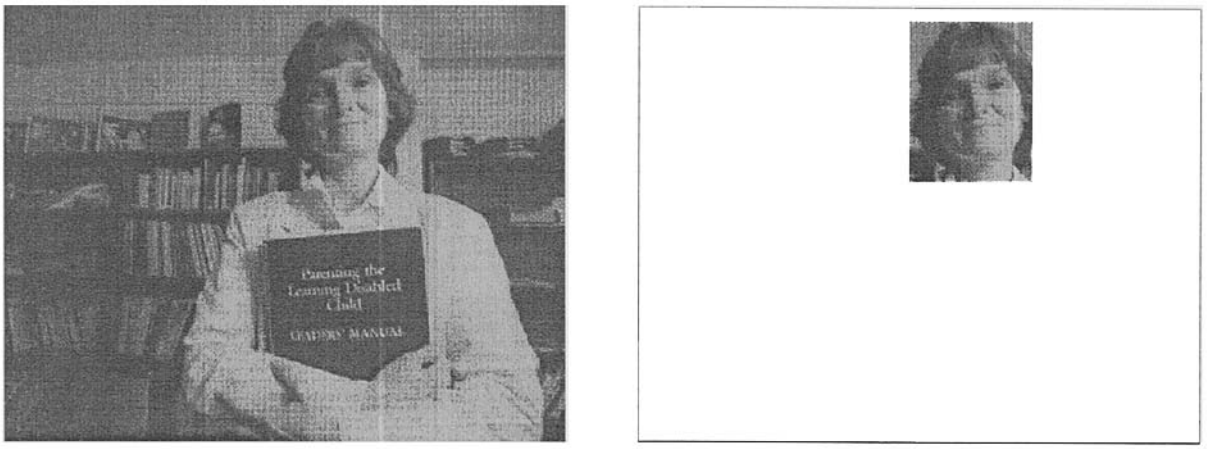


Figure 1. Objective: Frame human face(s) when the scene is cluttered with non-facial objects, (a) *Input*, a typical newspaper photograph, (b) *Output* of the face locator.

regions are verified by searching for face-specific features. Our work brings the *hypothesis generate and test* paradigm to bear upon the task of face location.

Section 2 gives a background discussion of the previous work in the area of locating faces in diverse scenes. Section 3 describes the model of a generic face and the details of actual implementation of all the algorithms that form part of the system. The feature selection process is studied from both a cognitive as well as a computational point of view. Section 4 is a report of the experiments performed to validate the approach. Section 5 is contains a brief conclusion.

2. Previous Work

Most research on human faces has focused on identifying a well-framed face, i.e., mug-shots. Moreover, the earlier work in face identification required human operator intervention. The operator would manually point to the features (eyes, mouth etc.) or describe them to aid the identification process. Early work on automating the feature extraction process for face identification was the work of Kanade (1973). He used a top-down control strategy directed by a generic model of expected feature characteristics. The face identification system calculated a set of facial parameters from a single face image and used a pattern classification technique to match the face from a known set of faces. The approach was purely statistical depending primarily on local histogram analysis.

The first effort in face location was reported by Sakai et al. (1969). They define the model of a human face in terms of several subtemplates corresponding to face specific features like the head line, eyes, mouth etc. Edges in the input photograph are matched against the subtemplates. The extent of match between subtemplates with similar spatial relations as that of the human face is used to predict the face location. Template matching techniques have the advantage of being simplistic. However, for the task of face location they prove to be inadequate. Variation in scale, pose, and shape cause direct template matching to be rather ineffective. Multiresolution/multiscale approaches have been proposed by many authors to achieve scale invariance.

A framework of templates and springs that can be used for general object recognition is described by Fischler and Elschlager (1973). They report experiments with human faces by describing a face as an object composed of several templates (e.g., eyes, mouth, nose, etc.) with a somewhat loose spatial relationships modeled by springs.

The few attempts in locating faces in scenes thus far, have assumed a benign background (Bromley, 1977; Lambert, 1987; Smith, 1966). The task of locating a generic human face in a cluttered background has been relatively unexplored. We have previously advocated the importance of the problem from an application point of view as well as from the perspective of general purpose recognition of natural objects (Govindaraju et al., 1989, 1990a). Significant effort was put into locating faces without *a priori* knowledge of the sizes of faces to be expected in the picture. Scale invariancy by either manual adjustment of templates or from distances between internal features (e.g., eyes) where

proportions and placement of features for a typical head were obtained from a drawing instruction manual (Lambert, 1987).

Use of neural networks for location of has been studied by many authors. Seitz and Lang (1991) use local orientation features instead edges. An approach similar to the Hough transform where feature matching is done by accumulating evidence in binary channels is used. Decisions are obtained in a neural network type architecture where non linear weighting and summation of the accumulated evidence is used for decision making. Vailliant et al. (1993b) use a two stage neural approach to the problem. The first stage roughly locates the possible regions, while the second stage does the actual location. The same authors (Vailliant et al. 1993a) also propose a multiresolution approach. These methods are tolerant to scale and pose variances to an extent, but their ability to handle multiple faces and extreme variations of scale is not clear. Also high clutter and very close location of faces will cause errors. Augusteijn and Skujca (1993) use texture of hair and skin as features for location, they use a Kohonen type network for the location. Use of texture as features restrict the minimum size of faces that can be located.

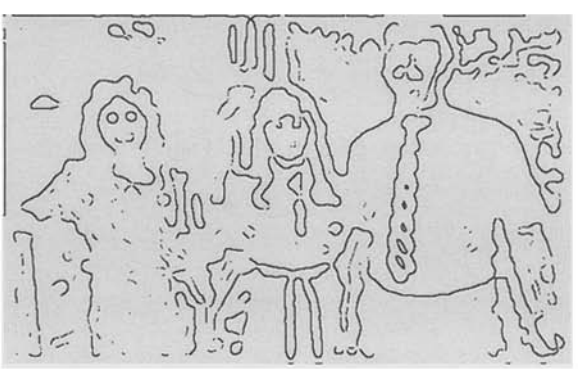
Deformable templates and an energy minimization paradigm to tackle the issue of scale are used by Yuille et al. (1988). However, their technique requires the template be placed in proximity of the object of interest before processing begins. Features of interest (e.g., eyes, mouth) are described by parameterized templates. An energy function is defined to link the edges, peaks and valleys in the image to corresponding parameters in the template. The template then interacts dynamically with the image, by altering its parameter values to achieve energy minimization. This results in self-deformation of the template to find a "best fit" in the image.

Recently, a near-real time system that performs both face location and identification was developed by Turk and Pentland (1991). Their system can track a subject's head, and then identify the person by comparing characteristics of the face to those of known individuals. They assume that faces in the images are upright. The experiments described deal with a single person in the image. Household and office environments are used for backgrounds. The system functions by projecting face images onto the feature space that spans the significant variations among known face images. The significant features are known as "eigenfaces" because they are the eigen vectors of the set of faces. The projection operation characterizes an individual face by a weighted sum of eigenface features.

3. System Implementation

Input to the face locator is a digitized newspaper photograph (Fig. 2(a)) The Marr-Hildreth edge operator is used to obtain an edge-image of the photograph (Fig. 2(b)). The edges are thinned by a classical thinning algorithm. Contours of objects with shapes and sizes that are unlikely to be parts of a human face are deleted by "filtering" procedures. For example, the contours corresponding to the tie of Mr. Barlow in Fig. 2(b) have a distinctive shape which is different from the contours of any possible human face. Such contours are filtered out (Fig. 2(c)). The edge detection and the

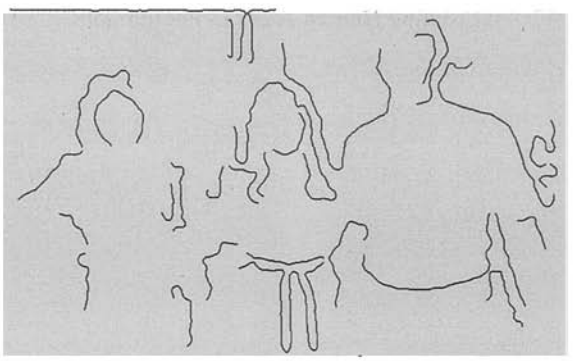
(a)



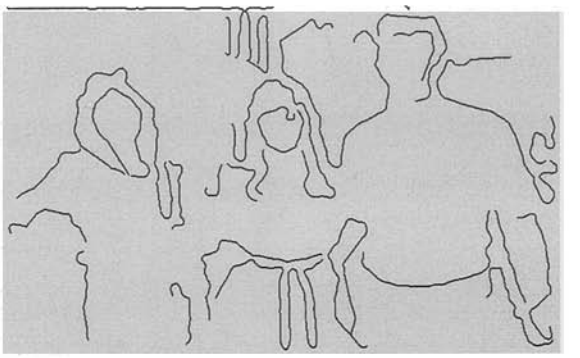
(b)

Figure 2. (a) Newspaper photograph with caption: *Bernard and Mai Barlow with her daughter Nguyen Thi Ngoc Linh in Leominister Massachusetts*, (b) Edge image obtained by the Marr-Hildreth operator ($\sigma = 5$) followed by a zero-crossings detector.

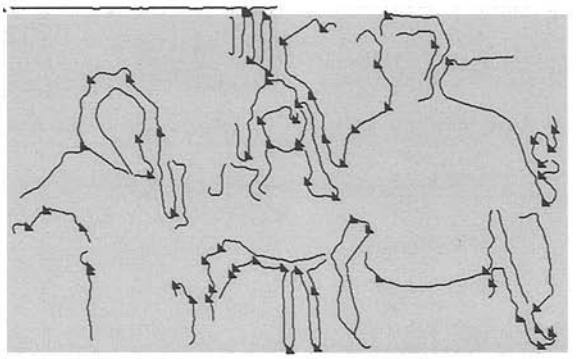
(*Continued on next page*)



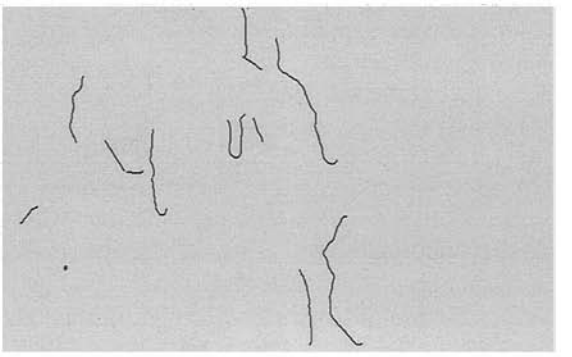
(c)



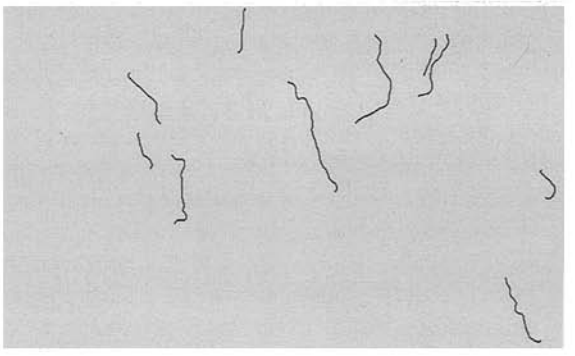
(d)



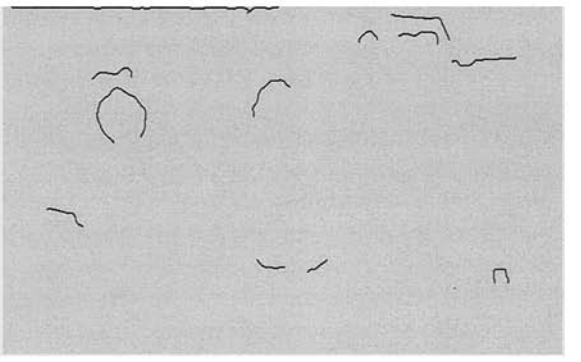
(e)



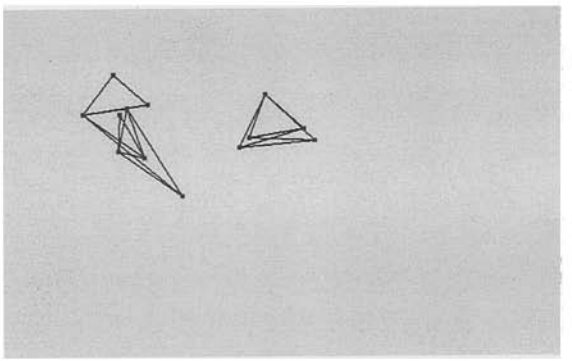
(f)



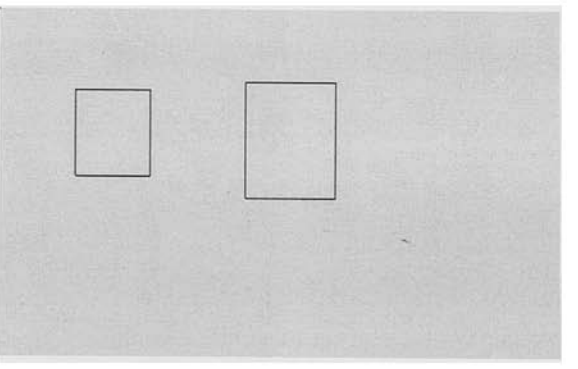
(g)



(h)



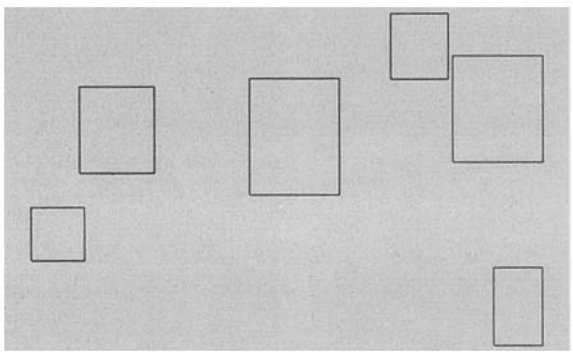
(i)



(j)

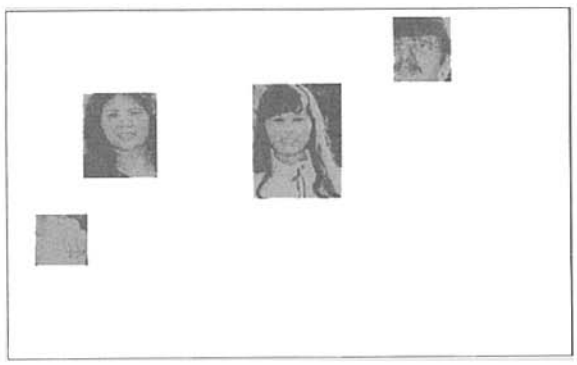
Figure 2. (*Continued*) (c) Contours get fragmented after filtering and thresholding operations are performed, (d) Fragmented contours are linked using (b) and (c). (e) Corners are marked to segment contours into features, (f) Feature curve labeled as $\mathcal{L}$. (g) Feature curves labeled as $\mathcal{R}$, (h) Feature curves labeled as $\mathcal{H}$. (i) Low cost cliques group the triplets $\mathcal{L}$, $\mathcal{R}$, $\mathcal{H}$, (j) Candidates hypothesized in the 1st pass.

(*Continued on next page*)



(k)

(l)



(m)

(n)

Figure 2. (Continued) (k) Candidates hypothesized in the 2nd pass, (l) Hypothesis verification phase must confirm 3 of the 6 candidates. (m) Moments-based filters reject 2 of the 3 false candidates, (n) Caption-aided verification process: The false candidate is rejected.

filtering process tend to fragment contours. Pairs of fragmented contours are linked based on the proximity of fragments and their relative orientations (Fig. 2(d)). Corners in the contours are detected to segment them into feature curves (Fig. 2(e)).

We have defined a generic face model for the hypothesis generation phase in terms of features based on the edges of the front view of a face (Fig. 3). $\mathcal{L}$, $\mathcal{H}$, $\mathcal{R}$ correspond to the left side, the hair-line and the right side of the front view of a face. The chin-curve is not used because the chin rarely shows up in the edge-data.

The relative sizes of the features are determined based on anthropometric literature which recommends using the *golden ratio*[2] for an ideal face (Farkas and Munro, 1987). Defining the relative sizes of features in terms of a ratio rather than absolute measurements ensures scale independence.

Although the model of the face is defined in terms of the frontal view of a face it is not limited to only such views. Side views of faces tend to have a similar shape. However, they do not adhere to the golden ratio. Therefore at the hypothesis generation stage, side views of faces are also matched, but with a lower score. It is expected that a more detailed analysis of face specific features and their positions will be required to discriminate between side-views and frontal views.

The features are then labeled by examining their geometric properties and the relative positions of other features in the neighborhood. Features that correspond to the left part of the face are labeled $\mathcal{L}$ (Fig. 2(f)); features that correspond to the right part of the face are labeled $\mathcal{R}$ (Fig. 2(g)); and features that correspond to the top of the face are labeled $\mathcal{H}$ (Fig. 2(h)). Locations of the centroids of the feature curves are embedded as nodes on a plane and a graph is constructed. Each node is attached with a list of attributes: the label of the feature curve, length, orientation, and area. Pairs of feature curves ($\{\mathcal{L}, \mathcal{R}\}$, $\{\mathcal{L}, \mathcal{H}\}$, $\{\mathcal{H}, \mathcal{R}\}$) are joined by edges if their attributes are compatible, i.e., if there is a possibility for the features to be parts of the same face.

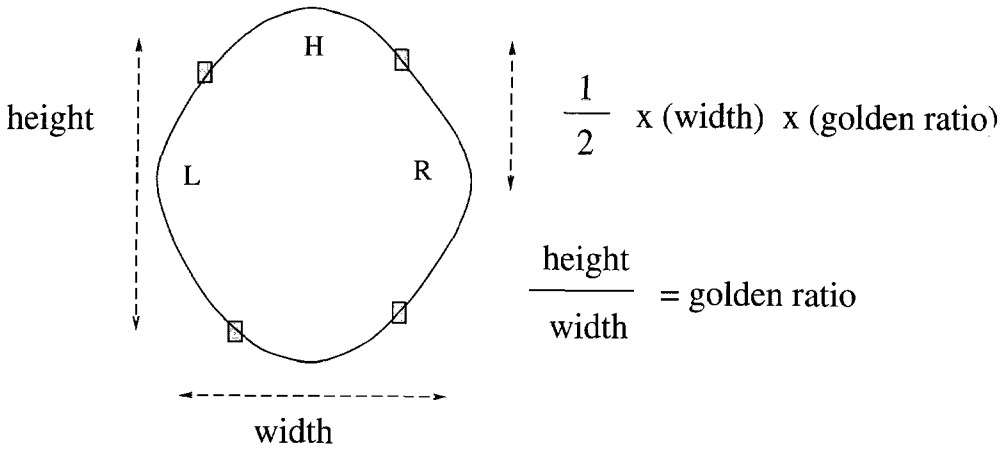


Figure 3. Model definition based on cognitive principles: Features (L, R, H) are selected for the hypothesis generation phase. The relative sizes of the feature curves is defined as a ratio to ensure independence.

Cliques[3] in the graphspace (triplets of $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$ are identified (Fig. 2(i)). The ratio of the pairs of features forming an edge is compared with the golden ratio and a cost is assigned to the edge. The cost of a clique is a function of the cost of the constituent edges. The cost reflects the distance between the instance of the shape of the hypothesized face in the image to the shape of an ideal face as described by a model. Cliques of low cost are selected to form the hypothesis for locations of people. (Fig. 2(j)).

Collateral information (from the caption) indicates that the system expects three people in the picture. Since, only two candidates were located (Fig. 2(j)), the entire process of hypothesis generation is repeated by relaxing some of the requirements specified in the definition of the model. For instance, instead of requiring the presence of triplets $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$ in order to hypothesize a face, it is deemed sufficient if just he feature pair $\{\mathcal{H}, \mathcal{L}\}$ or $\{\mathcal{H}, \mathcal{R}\}$ is present (Fig. 2(k)). Such a relaxation of requirements increases the number of hypothesized candidates as well as false alarms.

Given the hypothesized locations of people, some of which are false locations one must find the true locations and reject the others. Figure 2(l) shows 6 possible locations of people in the picture. The hypothesis verification stage must accept 3 of the 6 locations as true positions of people. The required number of true locations corresponds to the number of people mentioned by names in the caption of the picture.

Scope of this paper is limited to the hypothesis generation phase. Two different approaches can be adopted to solve the verification problem. The boundaries of the hypothesized locations can be used to retrieve gray-scale image data from those regions in the original photograph. Segments in the gray-scale image data can determine whether or not regions corresponding to eyes are present in that area. Filters can be developed that check if the hypothesized location is roughly symmetrical, if the location is purely homogeneous in its gray level distributions and so on.

Srihari (1991) describes a unique approach of rejecting false candidates in the context of newspaper photographs. Spatial constraints in the caption are used to predict the relative positions of positions of people in the picture. The system uses heuristic rules derived from codes followed in photojournalism (Arnold, 1969). Each name mentioned in the caption is associated with a hypothesized location. The locations which do not correspond to any of the people mentioned in the caption are rejected. The rule used in processing Fig. 2(m) is to prefer the group of candidates which are at approximately the same level. The false candidate in the picture is at a lower level than the other three and is hence rejected.

3.1. Feature Extraction

A string of operations is applied to the digitized newspaper photograph to obtain all instances of the features $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$. Following subsections describe the algorithms used in the feature extraction.

Edge Detection: Edges in picture are detected by convolving image with a Marr-Hildreth edge operator (1980) of mask size $31 \times 31 (\sigma = 5)$ followed by a zero crossings detection program.

Thinning: We adopt a classical algorithm (Pavlidis, 1982) to obtain a thinned image. The idea behind this algorithm is to identify all those pixels P_i which are essential in preserving the connectivity of the contours in the image. The pixels which are not essential are deleted as they do not belong to the skeleton of the contours.

Connected Components: In order to set up suitable data structures for further processing we must be able to access each connected component. Therefore each component must necessarily have an address which can serve as a key to accessing all its attributes.

Filters: Connected components are subjected to two filtering operations. First, all the components are made linear (no branches) by a spur removal process. Secondly, contours with non-face like properties (perfectly straight contour, contours with holes, etc.) are removed.

Spur Removal: Each connected component is a sparse graph representation of the structure of the component. It is usually a long spine with small branches and occasional loops hanging onto the spine. Consider the connected component C shown in Fig. 4. The first step of the filtering process is to obtain a linear structure from the skeletal graph, i.e., all pixels (nodes) in C must have a degree of 2 except at the terminal points where degree is 1. In the literature, this procedure is called "spur removal" (Walters, 1987) and is essentially the process of "chopping off" all branches from a central part in the graph which we will call the *spine*. In Fig. 4, AD is the spine of the graph and the branches terminating in C, D, E are *spurs*. The spur removal algorithm essentially examines all possible *paths* in the graph and picks the longest path with the minimum direction changes.

1. *Preservation of the smoothness of a contour*: When the trace of a path encounters a junction node with many alternative paths, the algorithm prefers the direction which causes the minimum change in the trend of the already established direction. In Fig. 4, starting from A, the algorithm chooses the path towards D at junction 3 rather than towards C because AD has fewer direction changes when compared to AC. The algorithm has a "memory" of the current direction trend so that every subsequent move is made to confirm to the established trend as far as possible.
2. *Inertia* is encoded in the algorithm so that small bumps cannot affect the direction trend. In Fig. 4, at junction 1, the path towards B causes lesser direction change than the path towards junction 2. However, since the bump at junction 1 is less than a pre-determined threshold it cannot influence the choice of alternatives.

From the above procedural definition, it is evident that the spine of a component is unique (the algorithm returns AD in Fig. 4). It should also be noted that the tracing procedure does not conform to the reflexive property. The path traced when starting from B terminates at A whereas the path starting from A terminates at D (Fig. 4).

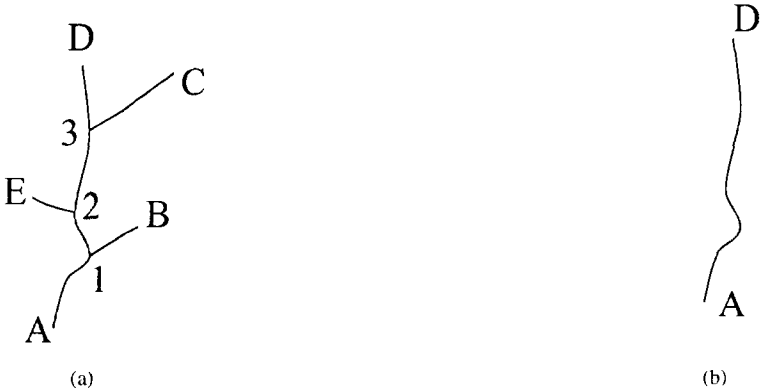


Figure 4. Spur removal: Given a skeletal graph (a), the spur removal procedure finds the central branch of the graph (b). AD is the spine of the graph; 1B, 2E, & 3C are spurs.

Algorithm
Input: Figure 3(b);
Output: Figure 3(a)
For all end points E_i in the graph
Trace a path starting from E_i
while preserving the "smoothness criterion" at each junction
Select the path which optimizes over length and minimum direction changes

Removal of Interfering Contours: Contours with the following properties are identified as interference: contours of very small length, contours with holes and wiggles (wavy shape like the tie of Mr. Barlow in Fig. 3(b)), and contours which are perfectly straight. Each contour is tested for the presence of any of these properties. A contour is "tiny" if the number of pixels in the component is less than a threshold ($thresh_{\text{size}}$). A hold is identified if all the pixels in contour have a degree of 2, i.e., there are no end points. A contour has wiggles if the length of the contour (in terms of number of pixels) is greater than the distance between the terminal points of the contour by a margin determined by a threshold ($thresh_{\text{gap}}$). A contour is assumed to be a perfect straight line if the least square error of a "best fit" line through the pixels in the contour is less than a threshold ($thresh_{\text{linear}}$).

Algorithm
Input: Figure 3(b);
Output: Figure 3(a)
Let
C be the connected component obtained after spur removal;
(A_x, A_y) & (B_x, B_y) be the extreme points of C;
length be the number of pixels in C;
then
chord of C is $\sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$;
height of C is the highest of its bounding box;
width of C is the width of its bounding box;
diagonal of C is $\sqrt{height^2 + width^2}$;
perimeter of C is $2 \times (height + width)$;
if (1st pass) **then**
Set ($thresh_{size}$, $thresh_{gap}$, $thresh_{linear}$) conservatively; -
else if (2nd pass) **then**
ReSet ($thresh_{size}$, $thresh_{gap}$, $thresh_{linear}$) liberally;
if ($length \leq thresh_{size}$)
then C_{tiny} = TRUE;
else C_{tiny} = FALSE;
if ($A_x \equiv B_x \wedge A_y \equiv B_y$)
then C_{hole} = TRUE;
else C_{hole} = FALSE;
if (C_{tiny} = FALSE)
if ($2 \times length \geq perimeter$)
if (C_{hole} = FALSE)
if ($chord \leq thresh_{gap} \wedge length \geq 2 \times chord$)
then C_{wiggle} = TRUE;
else C_{wiggle} = FALSE;
if (C_{wiggle} = FALSE)
if (for a straight line fit through the pixels of C *the least square error* $\leq thresh_{linear}$)
then C_{linear} = TRUE;
else C_{linear} = FALSE;
if ($C_{tiny} \vee C_{linear} \vee C_{hole} \vee C_{wiggle}$)
then *delete* component C;

Linking Contours: Intuitively, one would expect contours with endpoints in close proximity to be candidates for being linked together. Consider the contours C_1 and C_2 in Fig. 5(a). Let their lengths be L_1 and L_2 of which the lengths of the pieces of contour hanging over the corner points are l_1 and l_2. Let the end points of the contours be A and B which are r units of distance apart. If the line joining the endpoints A and B makes an angle θ with the horizontal then we can define the appropriateness of actually joining the endpoints by a force F

$$F \propto \frac{l_1 \times l_2}{r^2} \times \cos(\theta)$$

Using the inverse square law as a basis for deriving the above formula is no accident. As should be expected, the strength of the attraction force decreases as the gap that needs to be linked increases. It should be specially noted that F is a function of l and not of L as it is only the hanging part of C which contributes towards the attractive force. The importance of the $\cos(\theta)$ factor should be apparent from Fig. 5(b). The force of attractive between A and B is greater than that between A and C although they are closer. The overhanging part of the contour is very small in Fig. 5(c) to generate enough attractive force to link the gap between A and B.

Algorithm
Input: Figure 3(b) **and** Figure 3(a);
Output: Figure 3(b)
Find the corners in the contours in the image;
Evaluate the lengths of the overhanging parts;
For all pairs of endpoints (i, j) within a range ($r_{ij} \leq \tau$)
Compute the force of attraction F_{ij};
Sort the pairs in descending order of F_{ij};
(Begin linking pairs in the sorted order)

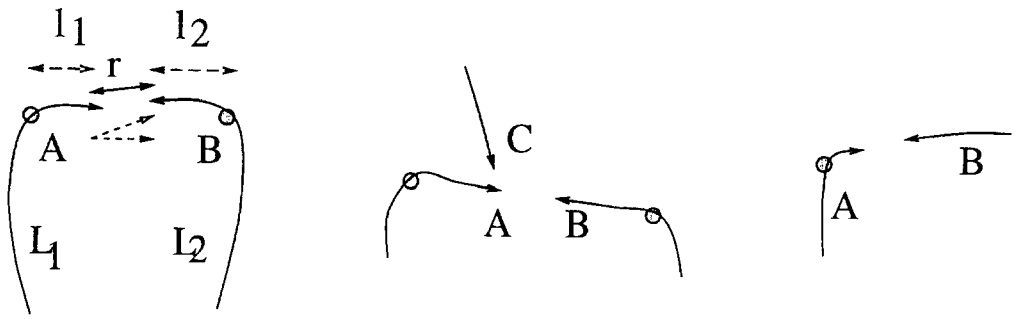


Figure 5. Linking algorithm: Based on proximity, direction and length.

if ($F_{ij} \geq thresh_{force}$) $\wedge$ (i & j are still unlinked)
then
Link endpoints i, j
Mark i & j as linked;

Corner Detection: Two different techniques of finding corners in chain-coded contours have been implemented. A union of the corners detected by each method is used. Freeman and Davis (1977) define a corner as an isolated discontinuity (local curvature) in the mean slope, its prominence (the cornerity of this point) being proportional to the length of the discontinuity-free regions to either side as well as the severity of the discontinuity.

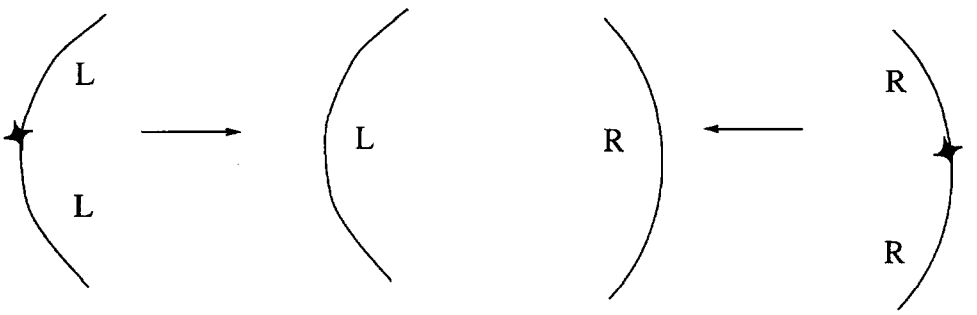
Method 1 (Medioni and Yasumoto, 1987): A parametric cubic B-spline is fit to the boundary curve and then the displacement between the original point and the interpolating spline is used to decide whether the point is a corner. Those points for which the displacement exceeds a threshold and the curvature is high are marked as corners.

Method 2 (Beus and Tiu, 1987): The second method is an improvement on the scheme reported in Freeman and Davis (1977). The prominence of a corner is the product of the length of the uniform chain sections to either side of a point, and the angle of the discontinuity at that particular point.

Assigning Labels to Feature Curves: A feature curve is the segment of contour flanked by end points or corner points on either side. At this stage we have access to all segments of contours which could potentially belong to a face. The next task is that of assigning labels to the feature curves as one of $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$. First, the geometric properties of the features (shape, direction of concavity, etc.) are used to determine the labels of the feature curves. However, the labels of some of the features cannot be determined just by the geometric properties. For instance, a feature which is approximately a vertical straight line can be either an $\mathcal{L}$ or an $\mathcal{R}$ because of the absence of any concavity. Such unlabeled features are marked with X. Often, if a neighbor of X is labeled as $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$, then the label of X can be uniquely determined. For example, if X has a $\mathcal{H}$ attached to its right side and the centroid of X is below the centroid of $\mathcal{H}$, then $X \equiv \mathcal{L}$ (**Rule 2**). A set of such rules, which constrain the label of a feature (X) by examining the labels of its neighbors, are used by the labeling algorithm in the second stage.

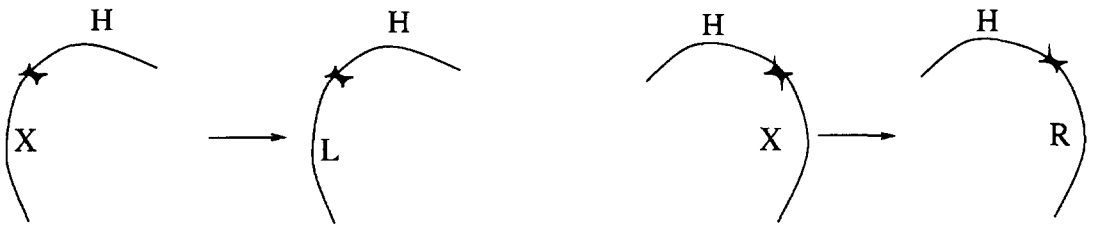
Algorithm
Input: Figure 3(a);
Output: Figure 3(b), Figure 3(a), **and** Figure 3(b);
Determine the direction of *concavity* of all feature curves;
if *concavity* faces to the left
then Label the feature with L
elseif *concavity* faces to the right
then Label the feature with R
elseif *concavity* faces downwards
then Label the feature with H
elseif *concavity* faces upwards
then Delete the feature
elseif *concavity* direction cannot be determined
then
Label the feature X
Apply Rule$_{1...6}$ as found suitable

Rule 1



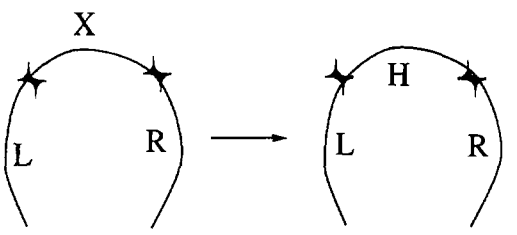
if identically labeled features are connected
then combine to form a single feature with the same label

Rule 2



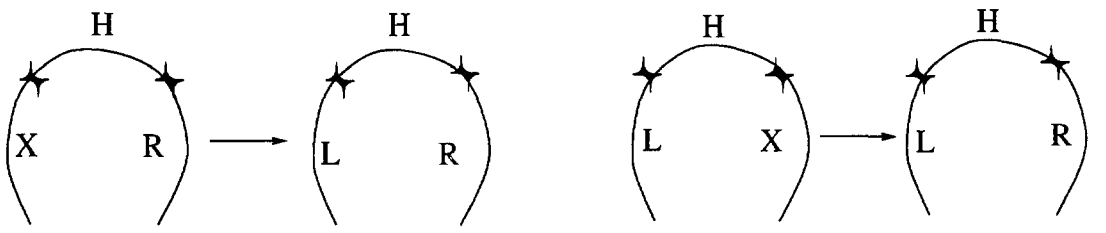
if a feature contour of unknown label (X) is vertical ($\theta \approx 90°$) **and** is connected to a feature contour labeled H above it
then label $X \equiv L$ if X is to the left of H and label $X \equiv R$ otherwise

Rule 3



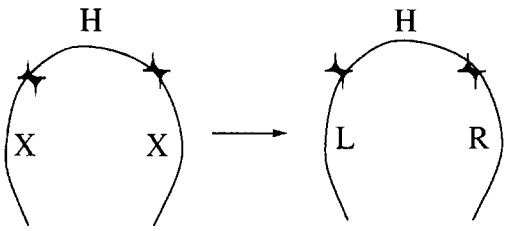
if a feature contour of unknown label (X) is horizontal ($\theta \approx 0°$) **and** is connected to a feature contour labeled L/R below it
then label $X \equiv H$

Rule 4



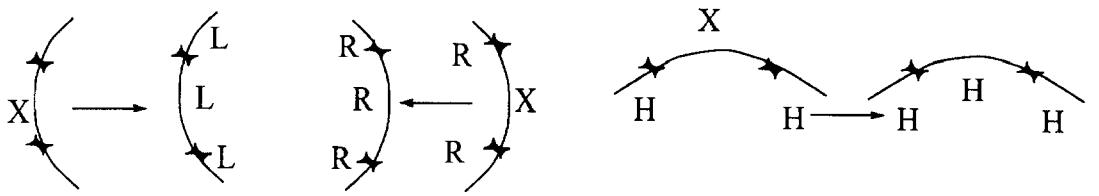
if a feature contour of unknown label (X) is vertical ($\theta \approx 90°$) **and** is connected to a feature contour labeled H above it
then label $X \equiv L$ if X is to the left of H and label $X \equiv R$ otherwise

Rule 5



Follows from repeated application of Rule 4

Rule 6



if a feature contour of unknown label (X)
is flanked between two features of the same label ($S \mid S = LVRVH$)
then label $X \equiv S$

3.2. *Representing the Extracted Feature Curves*

We represent each curve by a 4-tuplee $\{\vec{AB}, \mathcal{G}, \mathcal{A}, \ell\}$ (Fig. 6) and label it as one of $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$.

Let A and B bc thc cnd points of the curve;
$\vec{AB}$ be a vector corresponding to the chord of the curve;
θ be the angle made by $\vec{AB}$ and the horizontal;
ℓ be the length of the curve in pixels;
A be the area enclosed by the curve and the chord in pixels;

Our 4-tuple is a minor modification on the representation used by Cheng and Huang (1982). They argue that the attributes can be normalized so that they are invariant to scaling and rotation. Moreover, the attributes have the additive property, i.e., if two curves AB and BC are merged, then the resulting curve AC can be directly obtained from the attributes of AB and BC.

$\frac{l}{\mathcal{A}}$ represents the compactness of the curve. A straight line is the most compact curve. We expect $\mathcal{L}$ and $\mathcal{R}$ to have higher compactness than $\mathcal{H}$. $\mathcal{L}$ has majority of the area $\mathcal{A}$ on the left side of $\vec{AB}$; $\mathcal{R}$ has the majority on the right. We assume that the curve is localized at the centroid ($\mathcal{G}$) of its area (A). The scale of a feature curve is a functional of length of the curve (ℓ) and its area ($\mathcal{A}$).

3.3. *Feature Matching*

Figures 3(a), (b), and 3(a) show the features extracted from the image. However, not all the features extracted belong to faces. Matching the extracted features against the model (Fig. 3) allows one to identify those features which belong to faces. Matching the features against the model has two parts. First, matching the geometric shapes of the extracted features against the curves of an ideal face. Secondly, comparing the relative positions of the extracted tripleted triplets of features $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$ against the ideal relative positions derived from the model. Observing the *golden ratio rule*, for each feature curve in the image, the ideal position of its counterparts can be determined.

Cost Functions. In order to put the matching process in a mathematical framework, we have adopted a methodology of templates and springs (Fischler and Elschlager, 1973). This framework is well-suited to

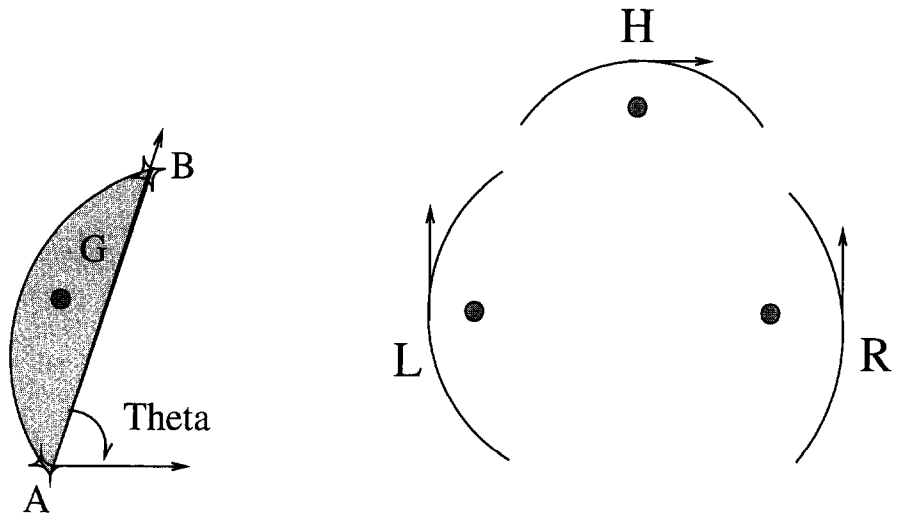


Figure 6. Feature representation: (a) representing curves with a 4-tuple: $\{\vec{AB}\ \mathcal{G}, \mathcal{A}, \ell\}$ and (b) labeling the curves as left, right, and hair, i.e., $\epsilon\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$.

quantify the "goodness" of correspondence between triplets in the image to the model.

Consider an embedding of the locations of all the features ($\mathcal{G}$) in a plane. Each of the locations is a node in the graph with a cost ($Cost_{\text{template}}$). Pairs of nodes with finite cost of association ($Cost_{\text{spring}}$) are joined with edges. Since there are three types of nodes ($\mathcal{L}$, $\mathcal{R}$, $\mathcal{H}$), the graph is 3-colorable. Since self-loops cannot occur, the only cycle in the graph is a triangle. It is also the biggest *clique* in the graph. Maximal cliques in the graph indicate the triplets that group together to form a candidate. Hence, the task of hypothesis generation is equivalent to finding cliques in the graph (Fig. 3(a)).

Consider a $\mathcal{L}$ feature curve located in the image at the position $\mathcal{G}_{\mathcal{L}}$ having a scale $\mathcal{S}_{\mathcal{L}}$ and an $\mathcal{R}$ feature curve located at $\mathcal{G}_{\mathcal{R}}$ with a scale of $\mathcal{S}_{\mathcal{R}}$ (Fig. 7). Ideally, the counterpart of $\mathcal{L}$ would be located at a distance of D ($\mathcal{S}_{\mathcal{L}}$, $golden_{\text{ratio}}$) from $\mathcal{G}_{\mathcal{L}}$ where D is a distance function such that the centroids of the counterpart and the image curve are in the same horizontal. However, the position of the $\mathcal{R}$ curve in the image may be at a different position ($\mathcal{G}_{\mathcal{R}}$). Similarly, the position of the $\mathcal{L}$ curve is not ideal from the point of view of the $\mathcal{R}$ curve. Therefore the cost of pairing the feature curves in the image is a function of the following two parameters:

$$d = \frac{|\mathcal{G}_{\mathcal{R}} - D(\mathcal{G}_{\mathcal{L}}, golden_{\text{ratio}})| + |D(\mathcal{G}_{\mathcal{R}}, golden_{\text{ratio}})|}{\mathcal{S}_{\mathcal{L}} + \mathcal{S}_{\mathcal{R}}} \quad \text{(Fig. 7(b))} \quad (1)$$

$$\alpha = \tan^{-1} \frac{|\mathcal{G}_{\mathcal{R}y} - \mathcal{G}_{\mathcal{L}y}|}{\mathcal{G}_{\mathcal{R}x} - \mathcal{G}_{\mathcal{L}x}|} \quad \text{(Fig. 7(b))} \quad (2)$$

(Note: suffixes x and y correspond to the x and y coordinates)

d and α can be similarly computed for pairing feature curves $\{\mathcal{L}, \mathcal{H}\}$ and $\{\mathcal{R}, \mathcal{H}\}$.

Algorithm
Input: Figure 3(b), Figure 3(a), and Figure 3(b)
Output: Figure 3(a)
Let Δ be an upper bound on the face-size;
For all curves c_i in the image **do** {
For a window $2\Delta \times 2\Delta$ centered at $\mathcal{G}$ of c_i **do**
{ **For all** pairs of curves (c_i, c_j) in the window **do** {
Edge-cost (c_i, c_j);
Reject all pairs of curves with cost $\geq$ a threshold, (*say*, τ);
Embed the surviving pairs into a *graph*;
Find all *cliques* (triangles):
($\bigcirc e^{\frac{3}{2}}$ complexity (Ciba and Nishizeki, 1985))
if two triangles share one or more nodes
then keep the clique of lower cost;
Rank the cliques by their cost; } } }

Clique-cost$(c_{\mathcal{L}}, c_{\mathcal{R}}, c_{\mathcal{H}})$
$Cost_{clique} = \Sigma_{edge \in clique} Cost_{edge}$
Let c_{scale} = length of curve c;
Let μ_{scale} = mean length of the curves in a clique;
For curves $c_{\mathcal{L}}$, $c_{\mathcal{R}}$, and $c_{\mathcal{H}}$
if $|c_{scale} - \mu_{scale}| \geq threshold$
then add penalty to $Cost_{clique}$;
if $C_{\mathcal{L}}$ is not connected to $C_{\mathcal{H}}$
then add a penalty cost to $Cost_{clique}$;
if $C_{\mathcal{L}}$ is not connected to $C_{\mathcal{H}}$
then add a penalty cost to $Cost_{clique}$;

Edge-cost (c_i, c_j)
$Cost_{template}(c_i) \propto \theta$; $Cost_{template}(c_j) \propto \theta$;
(θ is the mismatch between c_i and ideal curve, Figure 8)
$Cost_{spring}(c_i, c_j) \propto d, \alpha$ (Equation: 0);
Constrain-search (c_i, c_j);

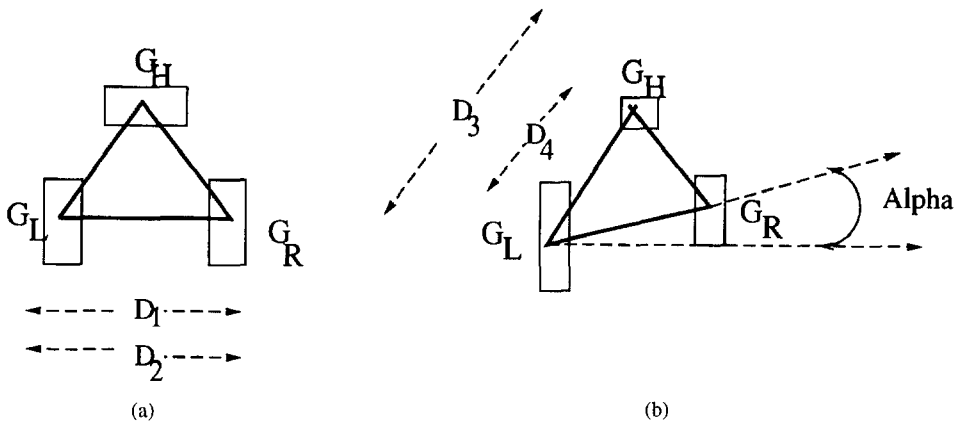


Figure 7. Configuration of templates (a) in an ideal face (Cost = 0); D_1 is the ideal distance from $G_{\mathcal{R}}$ to $G_{\mathcal{L}}$ using the $Golden_{\text{ratio}}$ and S_L as the ideal scale of the face. D_2 is the ideal distance from $G_{\mathcal{R}}$ to $G_{\mathcal{L}}$ using the $Golden_{\text{ratio}}$ and S_R as the ideal scale of the face. $d = \frac{D_1+D_2}{S_L+S_R}$. (b) Configuration of triplets in the image (finite cost). D_3 is the ideal distance from G_L to G_H using the $Golden_{\text{ratio}}$ and S_L as the ideal scale of the face. D_4 is the ideal distance from G_L to G_H using the $Golden_{\text{ratio}}$ and S_R as the ideal scale of the face. $d = \frac{D_3+D_4}{S_L+S_H}$

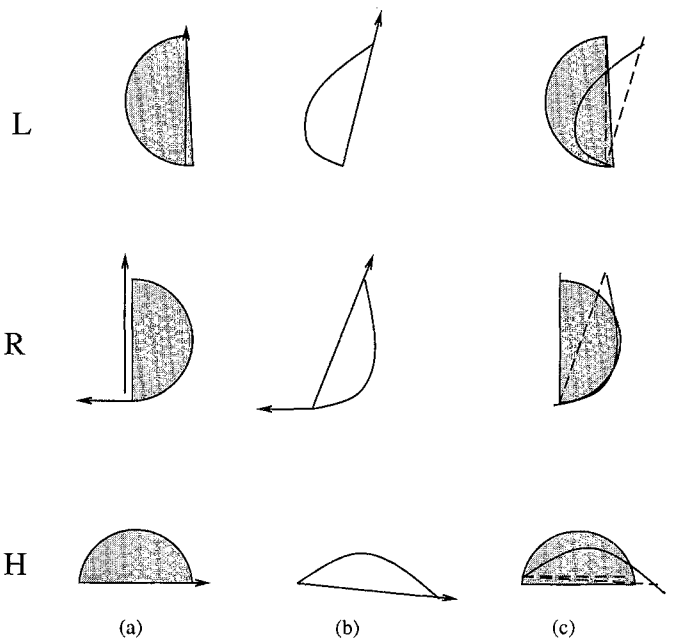


Figure 8. Costs of features curves: (a) ideally shaped model feature curves, (b) actual feature curves extracted from an image, (c) $Cost_{\text{template}} \propto$ mismatch between the ideal curves defined in the model and the actual curves present in the image.

if cost (c_i, c_j) is finite
then $CostEdge_{i,j} = Cost_{spring}(c_i, c_j) + \Sigma_{x=i,j} Cost_{template}(c_x)$;

Constrain-search(c_i, c_j)
Let the window $(2\Delta \times 2\Delta)$ be centered at c_i;
Let the vertical distance between (c_i, c_j) be v;
Let the horizontal distance between (c_i, c_j) be h;

if $(i, j) \equiv (\mathcal{L}, \mathcal{R})$, as in Figure 9
and $c_{\mathcal{L}}$ is left of $c_{\mathcal{R}}$ in the window
and $v \leq \Delta$ **and** $h \leq \frac{\Delta}{2}$
then cost(c_i, c_j) is finite **else** infinite;

if $(i, j) \equiv (\mathcal{H}, \mathcal{L})$ or $(i, j) \equiv (\mathcal{H}, \mathcal{R})$ as in Figure 9
and $c_{\mathcal{H}}$ is above $(\mathcal{L}$ or $\mathcal{R})$
and $v \leq \Delta$ **and** $h \leq \frac{\Delta}{2}$
then cost(c_i, c_j) is finite **else** infinite;

Pairs of curves of different types $(\mathcal{L}, \mathcal{R}, \mathcal{H})$ are connected by 'springs' represented by 'Edge-cost'. This 'Edge-cost' is finite if and only if it reamins in a finite neighborhood of each other as shown in Fig. 9. Groups of three contours are formed and the total cost 'Clique cost' is determined based on the 'Edge-costs'

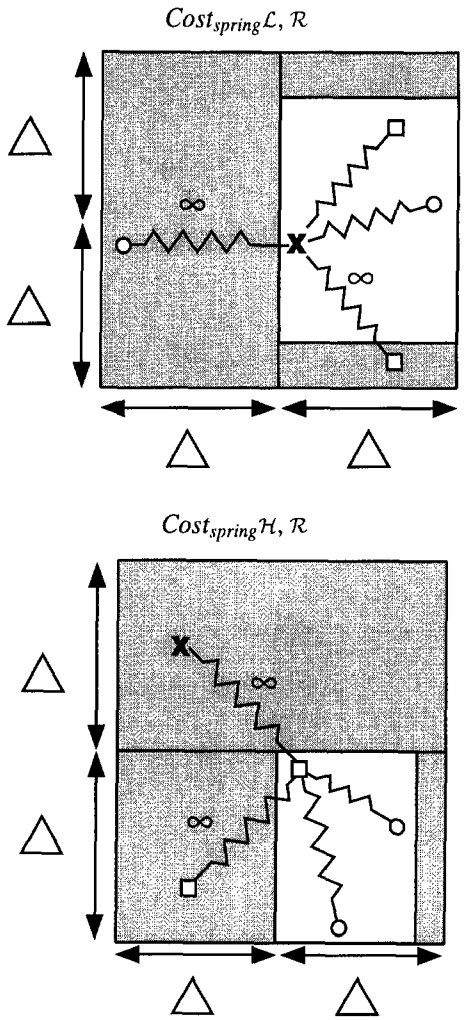


Figure 9. Plausible pairings of feature curves: Constrain search to finite cost neighborhoods. □: position of a $\mathcal{H}$ feature curve; ○: position of a $\mathcal{R}$ feature curve; ×: position of a $\mathcal{L}$ feature curve.

This 'Clique cost' is used as the basis for deciding possible face locations.

4. Experiments

The training set was digitized at 200 pixels per inch (ppi) using an Eikonix camera and subsequently reduced to 100 ppi to decrease processing time. The test set was digitized at 100 ppi by a Microtex scanner. The face locator is implemented entirely in software (C programs) and runs on a SUN Sparc workstation.

So as to avoid subjectivity, we shall consider all faces mentioned by the caption as identifiable.

We shall assume our program has "hit" a target if the position of the face is marked by a frame which encloses at least the eyes of the face but is no larger than twice the area of the face. If we consider the unit of evaluation as an entire picture we need further classifications. Let us define the performance measures for the entire picture as in Table 1: success (SU), partial success (PS), and error (ER).

Table 1. Performance code reference.

Performance code	Definition	Hits
SU: Success	All identifiable faces in the image are 'hit'	$=n$
PS: Partial success	At least half the identifiable faces are 'hit'	$\geq \lfloor \frac{n}{2} \rfloor$
ER: Error	Less than half the identifiable faces are 'hit'	$< \lfloor \frac{n}{2} \rfloor$

As the probability of a locating a face increases so does the probability of false alarms. When the system allows fewer false alarms (and hits), the thresholds of all parameters contributing towards the hypothesis of a face are *constrained* and when the system aims for larger number of hits (and false alarms) the parameters are *relaxed.*

An example of relaxed and constrained parameter used in our system is the following. When the system wishes to generate as few false alarms as possible, our model requires that all three sides of an object be detected $\{\mathcal{L}, \mathcal{H}, \mathcal{R}\}$ in order to respond with a S. When the system, wishes to hit as many faces as possible, our model allows the detection of just two sides of an object as a sufficient requirement to hypothesize a face. The parameter representing the required number of sides has a value of 3 in the *constrained* case and 2 in the *relaxed* case.

We have tested our system on 3 different criterion of decision making which we call operating points.

1. P1: The system makes decisions with *constrained* set of parameters.
2. P2: The system makes decisions with *relaxed* set of parameters.
3. P3: The system works in two passes. In the first pass (pass 1) constrained parameters are adopted. If all the faces in the picture are not hit (number of misses ≥ 1), then the system is run again with the relaxed parameters (pass 2).

The number of false alarms is least in P1 and the number of hits is highest in P2 and P3. However, the number of false alarms of P3 is less than that of P2, because P3 relaxes the parameters only when necessary. Choice of the operating point should be guided

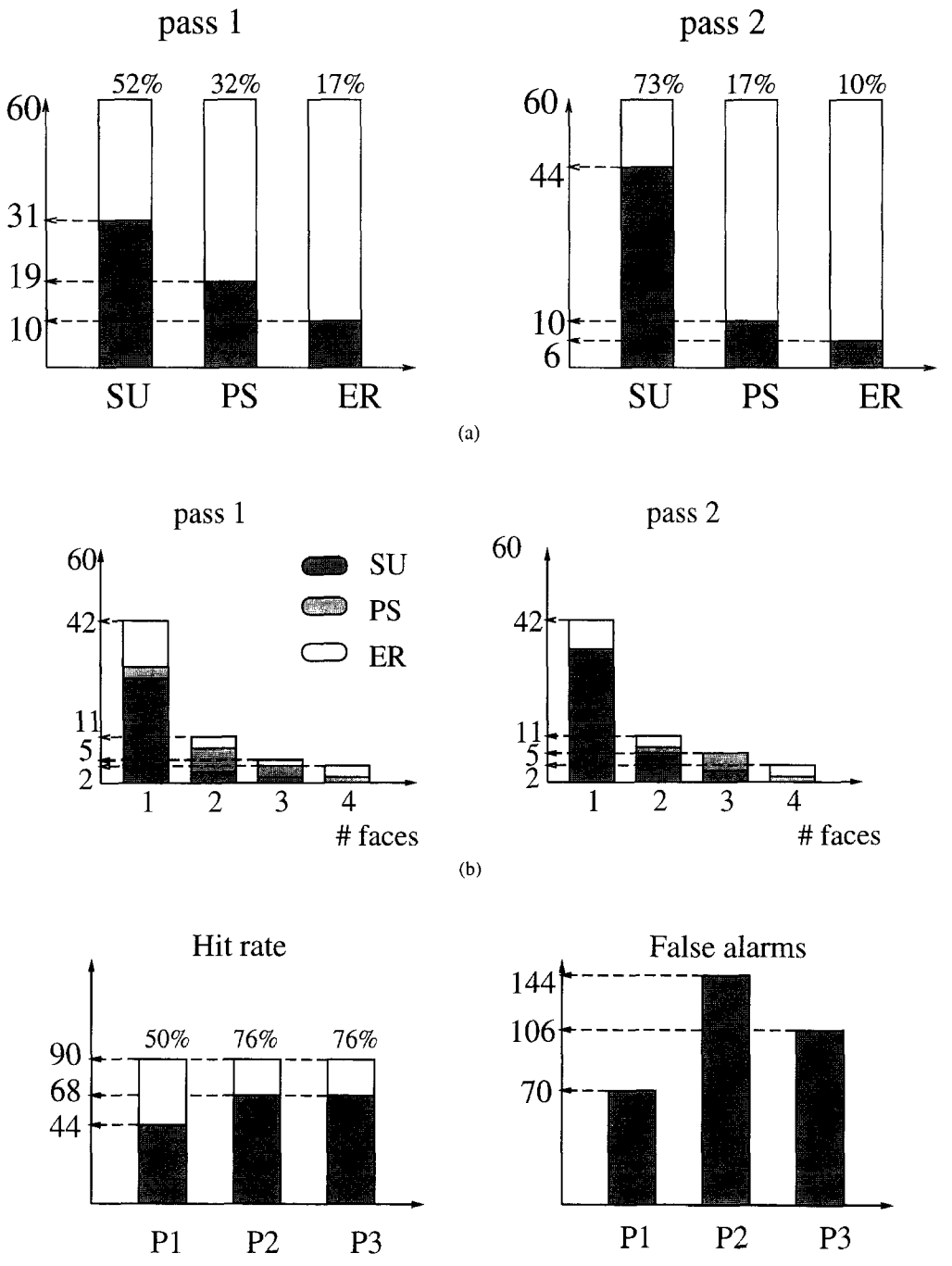


Figure 10. Analysis of results from the TEST SETS: (a) performance on each training image, (b) performance on images with $1 \cdots 4$ faces, (c) overall performance in terms of "hits" and "false alarms"; $P_1 \cdots 3$ are the plausible operating points; P_3 gives the best performance. SU, PS and ER refer to Success, Partial Success and Error—Table 1.

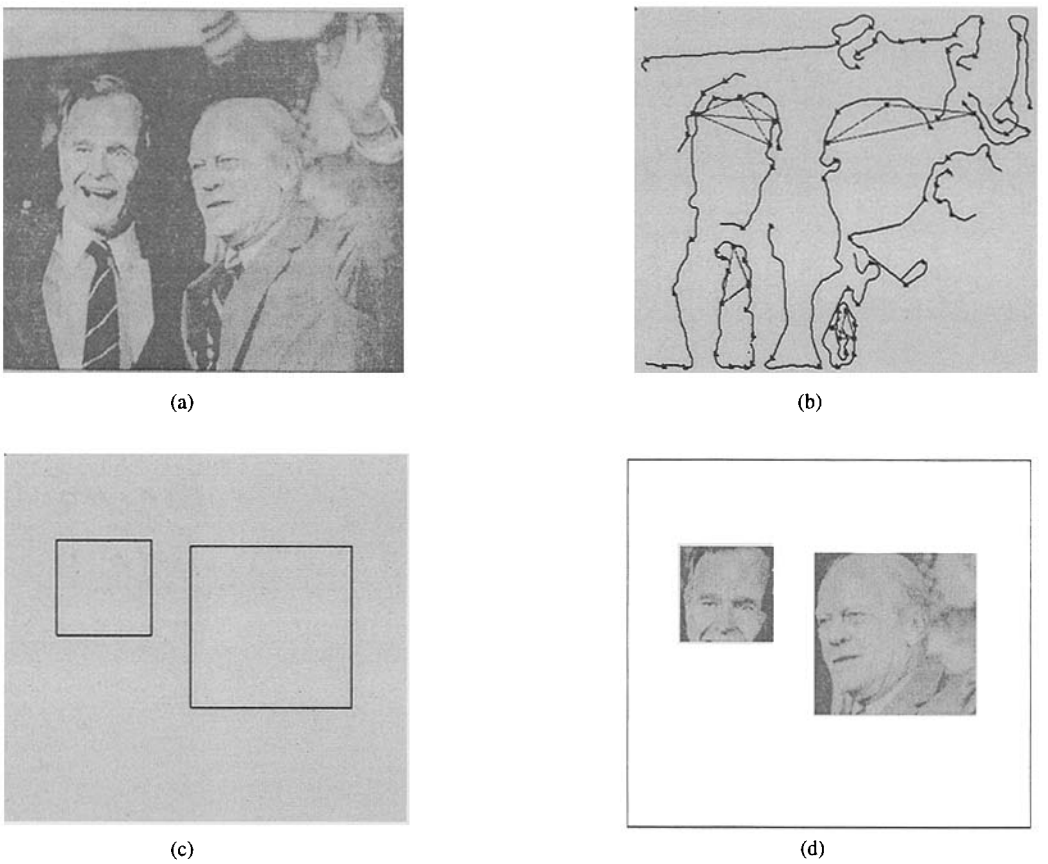


Figure 11. (a) Bush-Ford image, (b) Groups created after two passes, (c) Candidates hypothesized after two passes, (d) Located faces.

by the application requirements (tolerance of misses and false alarms). If the hypothesis verification programs are known to be accurate then more false alarms can be tolerated at the hypothesis generation stage. If the penalty of false alarms overwhelms the penalty of misses, then a more conservative set of parameters should be used (P1).

The feedback after the first pass is critical. The system resorts to the second pass whenever the number of candidates generated are less than the required number of faces (obtained from the caption). Calling the system a second time increases the computational time but reduces the number of false alarms while keeping the number of hits constant.

Out of 60 test pictures, the system located all the faces featuring in 44 of the pictures (73%). There were 90 faces among the 60 pictures (on average 1.5 faces per picture). The system located 68 of the faces (76%). On average, $\approx$2 false alarms were generated per picture and $\approx$1 false alarm per face. If a picture had a single face, our system was successful in locating the face 83% of the time. Details of the test in various scenarios are illustrated in Fig. 10.

Table 2 shows all the causes of failure we could compile from visual inspection of the output of the system. There are two types of error sources. Those that are inherent in the quality of photograph and those

Table 2. Sources of error.

Code	Error source	Detection
CT	Contrast	Missing edges
CL	Clutter	High edge density
ED	Edges	Spurious or missing edges
VI	Non-frontal face views	Crooked face contours
CO	Corner detection	Missing or misplaced corners
CC	Cost evaluation of cliques	Missing boxes inspite of cliques
FI	Filtering	Small contours and blobs removed
LA	Labeling contours	Implausible feature grouping
LI	Linking contours	Incorrect linking

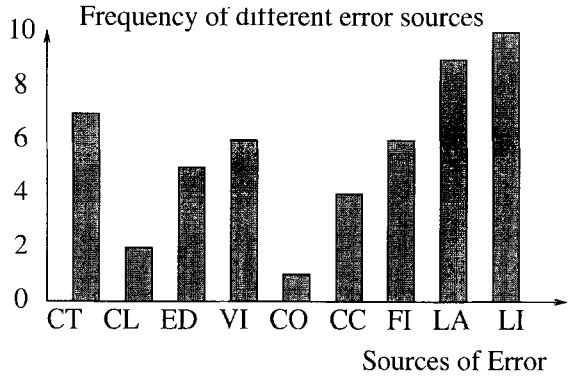


Figure 12. Sources of error in the TEST SET.

committed by our algorithms. The error sources related to the quality of pictures themselves are poor contrast of faces against the background, extreme background clutter, and people not facing squarely into the camera. Poor contrast often manifests as broken or missing edge data, while clutter in the picture results in a very high density of edges. Among the algorithm related errors, corner detection sometimes misses corners while picking spurious corners. The cost of cliques is evaluated using anthropometric measurements and involves empirical setting of thresholds at various levels which can potentially lead to error. The system uses subjective filters based on size, linearity, shape, etc. which can cause error. All contours are labeled as a $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$ based on their shape and concavity which is a process prone to errors because of the difficulty involved in accurately estimating the concavities. Table 2 shows the sources of error and the effect each error has on the data to be processed by the face locating programs. Figure 12 illustrates the frequency of different error sources in the test set. From the algorithmic stand point, labeling contours (LA) and linking fragmented contours (LI) cause a majority of the errors.

Figure 11 shows another example of the locator performance. Here it can been seen that to locate one of the faces, a contour other than one belonging to the face is used. In the two pass algorithm used, in the second pass, even if only two of the contours are present, a possible face candidate is hypothesized. Therefore frequently faces are located with only two of three contours, with the third one not being from the face itself or being entirely absent.

The system was tested on 50 regular home photographs (close-ups) of people as well with similar results. The system performed with a success rate of about 70%.

5. Conclusions

The face locator described here helps segmenting faces from general scences. Edge contours are used as the basic features. Edges located by the Marr-Hildreth detector are filtered and cleaned to obtain contours. The contours are labeled as left, right and head curves based on their characteristics. These contours are connected by 'springs' which are represented by the 'edge costs'. Groups of three curves $\{\mathcal{L}, \mathcal{R}, \mathcal{H}\}$ are made and a cost function evaluated. This cost function is used the basis to decide which of the groups represents a possible face candidate.

This locator serves as the first step towards characterization and indentification of human faces in a cluttered environment. This method is found reliable and the false alarm rate is low. A method for making use of collateral information is illustrated. The method explained here can serve as the basis for further research, work needs to be done in refining the location of faces by deriving a more precise fit than the ones obtained.

Acknowledgments

I would like to thank Professors S.N. Srihari, D.B. Sher, and D. Walters for their valuable guidance during the course of this project. Dr. R. Wildes (David Sarnoff Research Labs) helped me formulate the feature matching subtask in a mathematical framework. Dr. R. Gaborski (Eastman Kodak Company) gave valuable suggestions and helped motivate the research from a practical viewpoint. Finally, I am extremely grateful to M. Venkatraman, Research Scientist at CEDAR, who provided assistance in literature survey and suggested several improvements to bring the paper to its final form.

Notes

1. Pose refers to the view of a face captured by the retina.
2. Golden ratio: $\frac{\text{height}}{\text{width}} \equiv \frac{1+\sqrt{5}}{2}$, aesthetically proportioned rectangle used by artists.
3. Clique is a maximally connected (sub)graph. Every node in the (sub)graph is connected to every other node.

References

Arnold, E.C. 1969. *Modern Newspaper Design*. Harper and Row: New York, NY.

Augusteijn, M.F. and Skujca, T.L. 1993. Identification of human faces through texture-based feature recognition and neural network technology. *1993 IEEE Conference on Neural Networks*, 392–398.

Beus, H.L. and Tiu, S.S. 1987. An improved corner detection algorithm based on chain-coded plane curves. *Pattern Recognition*, 20:291–296.

Bromley, L.K. 1977. Computer-aided processing techniques for usage in real-time image evaluation. Master's Thesis, University of Houston.

Cheng, J.K. and Huang, T.S. 1982. Recognition of curvilinear objects by matching relational structures. In *Proc. Pattern Recognition and Image Processing*, pp. 343–348.

Ciba, N. and Nishizeki, T. 1985. Abrocity and subgraph listing algorithms. *SIAM J. of Computing*, 14(1):210–223.

Farkas, L.G. and Munro, I.R. 1987. *Anthropometric Facial Proportions in Medicine*. Charles C. Thomas: Springfield, USA.

Fischler, M.A. and Elschlager, R.A. 1973. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, c-22(1).

Freeman, H. and Davis, L.S. 1977. A corner finding algorithm for chain-coded curves. *IEEE Trans. Comput.* 26:297–303.

Govindaraju, V., Srihari, S.N., and Sher, D.B. 1992. A computational model for face location based on cognitive principles. In *Proc. of AAAI-92*, San Jose, CA, pp. 350–355.

Govindaraju, V., Sher, D.B., and Srihari, S.N. 1990. A computational model for face location. In *Proc. of IEEE-CS Third Int. Conference on Computer Vision*, Osaka, Japan, pp. 718–721.

Govindaraju, V. and Srihari, R.K. 1990. Automatic face recognition in news photo database. *Advanced Imaging*, 5(11):22–26.

Govindaraju, V., Sher, D.B., Srihari, N., and Srihari, S.N. 1989. Locating human faces in newspaper photographs. In *Proc. of IEEE-CS Conf. Computer Vision and Pattern Recognition*, San Diego, CA, pp. 278–285.

Govindaraju, V., Lam, S., Niyogi, D., Sher, D.B., Srihari, R., Srihari, S.N., and Lam, D. Newspaper Images Understanding. *Lecture Notes in Artificial Intelligence*, Vol. 444, pp. 375–386, J. Siekmann (Ed.), Springer Verlag, New York, NY.

Kanade, T. 1973. *Picture Processing System by Computer Complex and Recognition of Human Faces*. Department of Information Science, Kyoto University.

Lambert, L.C. 1987. Evaluation and enhancement of the AFIT autonomous face recognition machine. Master's Thesis. Air Force Institute of Technology.

Marr, D. and Hildreth, E. 1980. Theroy of edge detection. *Proc. of the Royal Society of London*, 207:187–217.

Medioni, G. and Yasumoto, Y. 1987. Corner detection and curve representation using cubic B-splines. *CVGIP*, 39:267–278.

Pavlidis, T. 1982. *Graphics and Image Processing*. Computer Science Press, 1803. Research Boulevard, Rockville, MD.

Sakai, T., Nagao, M., and Fujibayashi, S. 1969. Line extraction and pattern detection in a photograph. *Pattern Recognition*, 2:233–248.

Seitz, P. and Lang, G.K. 1991. Using local orientation and heirarchical spatial feature matching for the robust recognition of objects. *VCIP, Proceeding of SPIE*, 252–259.

Smith, E.J. 1986. Development of an autonomous face recognition machine. Master's Thesis, Air Force Institute of Technology.

Srihari, R.K. 1991. *Extracting Visual Information From Text: Using Captions to Label Faces in Newspaper Photographs*. Ph.D. Thesis, State University of New York at Buffalo.

Turk, M. and Pentland, A. 1991. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Vailliant, R., Monrocq, C., and Le Cun, Y. 1993. Original approach for the location of objects in images. *3rd Internation Conference on Artificial Neural Networks*, 372:26–29.

Vailliant, R., Monrocq, C., and Le Cun, Y. 1993. Location of faces in images. *Review Technique Thomson*, 25(1):23–40.

Walters, D. 1987. Spur removal in ρ-space. In *Personal Communication*.

Yuille, A., Cohen, D., and Hallinan, P. 1988. Facial feature extraction by deformable templates. Technical Report 88-2, Harvard Robotics Laboratory.