Fifth Intl. Conf. on Comp. Vision, Cambridge, MA (June 1995)

Finding Faces in Cluttered Scenes using Random Labeled Graph Matching

T.K. Leung[†‡], M.C. Burl[†], and P. Perona[†§]

† California Institute of Technology, 116-81, Pasadena, CA 91125, USA

‡ University of California at Berkeley, Computer Science Division, Berkeley, CA 94720, USA

§ Università di Padova, Italy

leungt@eecs.berkeley.edu, {burl,perona}@systems.caltech.edu

Abstract

An algorithm for locating quasi-frontal views of human faces in cluttered scenes is presented. The algorithm works by coupling a set of local feature detectors with a statistical model of the mutual distances between facial features; it is invariant with respect to translation, rotation (in the plane), and scale and can handle partial occlusions of the face. On a challenging database with complicated and varied backgrounds, the algorithm achieved a correct localization rate of 95% *in images where the face appeared quasi-frontally.*

1 Introduction

The problem of face recognition has received considerable attention from the computer vision community, and a number of techniques have been proposed in the literature [3, 11, 12, 13, 14, 16, 17, 19]. However, in most of these studies the face was in a benign environment from which it could easily be extracted, or it was assumed to have been pre-segmented. For any of these recognition algorithms to work in a general setting, we need a system that can reliably locate faces in cluttered scenes and with occlusions.

Recent studies have begun to address the problem of face localization. Burel and Carel [4] proposed a method based on multi-resolution analysis and learning from examples (multi-layer perceptron) to search for faces in an image. Yang and Huang [18] have described a system that locates faces using a hierarchical knowledge-based approach. Burt [6] has built a hardware system that works in semi-controlled environments. All of these systems, however, suffer from one or more of the following problems: (1) they do not work well with cluttered scenes, (2) they do not work if the face is partially occluded, or (3) they do not easily generalize to the problem of finding faces from different subjects.

In this paper, we propose a solution to the problem of face detection and localization using random graph matching. Graph matching has previously been used by Amit [1] for aligning X-ray images of hands; however, in his scheme the graphs are restricted to be of a special form (triangulated) in order to reduce the computational complexity; also, potential matches are scored based on an ad hoc energy function. We improve upon this work in three respects: (1) we use a rigorous probabilistic model to score potential matches, (2) we are able to explicitly handle missing features, and (3) our algorithm is more general since the object model may be described by *any* graph but the amount of computation is still reasonable.

In the initial step of our algorithm, a set of local feature detectors is applied to the image to identify candidate locations for facial features, such as the eyes, nose, and nostrils. Since the feature detectors are not perfectly reliable, the spatial arrangement of the features must also be used to localize the face.

The arrangement of facial features can be viewed as a random graph in which the nodes correspond to the features and the arc lengths correspond to the distances between the features. Since different people have different distances between their features, the arc lengths are modeled as a random vector drawn from a joint probability distribution. (We describe an alternative approach for modeling faces in [5]). The face localization task, therefore, corresponds to the problem of *random graph matching*; however, the statistical structure of our graphs is exploited to yield a computationally feasible algorithm.

2 Feature Detection

The initial step in our face localization algorithm is to identify candidate locations for various facial features. Although many methods are available for detecting features, in our experiments we have elected to use a technique based on matching descriptors produced by multi-orientation, multi-scale Gaussian derivative filters. Variations of this approach have been used successfully before, e.g., by Jones and Malik [10] in their stereo matching algorithm.

The incoming image is first convolved with a set of Gaussian derivative filters at different orientations and scales. The filtering is performed on a pyramid to reduce the computation time. The vector of filter responses at a particular spatial location $\mathbf{R}(x, y)$ serves as a description of the local image brightness. Candidates for the i^{th} facial feature are found by matching $\mathbf{R}(x, y)$ against a template (or prototype) vector of responses $\mathbf{P}_i$. The goodness of match $Q_i(x, y)$ is equal to the cosine of the angle between $\boldsymbol{P}_i$ and $\boldsymbol{R}(x, y)$; however, we also require that $\boldsymbol{P}_i$ and $\boldsymbol{R}(x, y)$ have similar

lengths. That is,

$$l_i(x,y) = \frac{|(|\boldsymbol{P}_i| - |\boldsymbol{R}(x,y)|)|}{|\boldsymbol{P}_i|} \quad (1)$$

$$Q_i(x,y) = \begin{cases} \frac{\boldsymbol{P}_i{}^T \boldsymbol{R}_{(x,y)}}{|\boldsymbol{P}_i||\boldsymbol{R}_{(x,y)}|} & \text{if } l_i(x,y) < \tau_0 \\ -1 & \text{otherwise} \end{cases} \quad (2)$$

A detection is declared at the point $(\hat{x}_i, \hat{y}_i)$ if $Q_i(\hat{x}_i, \hat{y}_i)$ is a local maximum and exceeds a threshold τ_{th}.

We can systematically look at the performance of each detector using receiver operating characteristics (ROC curves) [15]. Figure 1 shows the trade-off between the probability of detection and the number of false alarms per image for the first 150 images of the Lab Sequence (see Section 4.1 for a description of this database). Each plot shows the performance for a different feature. The ROC curves are implicitly parameterized by the threshold τ_{th} for values ranging from 0.50 to 1.00 in increments of 0.01. For τ_{th} close to 1, there are not many false alarms, but the true features are missed more frequently. For lower values of τ_{th}, the true features are detected, but there are many more false alarms.

From these curves, it is apparent that the features cannot be located with perfect reliability (100% probability of detection, 0 false alarms). Although this result could be blamed on the particular choice we have made for the detectors, we believe that it is almost certainly a property of feature detectors in general — *the local brightness information alone is not sufficient to insure perfect performance.* In the next section, we show how feature detectors can be coupled with a statistical model of the spatial arrangement of the features to improve the overall robustness of the system.

3 Graph Matching

Because of the high variability in appearance of the facial features (e.g., due to differences between subjects, viewpoint, or illumination), local detectors are not able to identify the feature locations with sufficient reliability. However, the configuration of the features can help us find the location of the face, since we know the features cannot appear in arbitrary arrangements. For example, given the positions of the two eyes, we know that the nose and the mouth can only lie within specific areas of the image.

To enforce the proper arrangement of features, constellations are formed from the pools of candidate locations and scored based on how face-like they appear. Finding the best constellation can be viewed as a random graph matching problem in which the nodes of the graph correspond to features on the face and the arcs (edges) represent the distances between the different features. With no heuristics, the complexity of matching a full graph is M^N, where N is the number of nodes in the template graph and M is the number of candidate points in the incoming image. For a large set of candidate points or a large number of nodes, this task is not computationally feasible. Therefore, full graph matching without heuristics can only be used for small values of M and N.

In the calculation above, we assumed that the candidate points were unlabeled. With labeled points, the situation is somewhat improved. The complexity of labeled graph matching is given by $\prod M_i$ from $i = 1$ to N, where M_i is the number of candidate points for feature i. The feature detection method discussed in Section 2 typically produces a small number of candidates for each feature (10-20 on average), so brute-force matching is feasible in this case. However, we will see later (in Section 3.2) that the graph matching complexity can be reduced even further by exploiting the statistical structure of our graphs.

3.1 Probability Model

A first hurdle is to choose a probabilistic model for the constellations. A general rule of thumb is that anthropometric data are jointly Gaussian if the specimens belong to the same population. Therefore, we will assume that the distances between the facial features are jointly Gaussian-distributed with some mean and covariance *provided the faces are normalized for scale.* Although the Gaussian density is clearly not "right" (distances have zero probability of taking on negative values), it is convenient analytically and appears to model well the variations present in human faces.

Let $\boldsymbol{X}$ be a random vector of mutual distances that *has not* been normalized for scale. We will refer to such a vector as a *natural vector of mutual distances.* Now, based on the magnitude of the distances in the components of $\boldsymbol{X}$, an overall scale factor $\hat{\lambda}(\boldsymbol{X})$ can be estimated and used to normalize $\boldsymbol{X}$. The resulting vector $\boldsymbol{L} = \frac{\boldsymbol{X}}{\hat{\lambda}(\boldsymbol{X})}$ will be referred to as a *scale-normalized vector of mutual distances.* Our basic assumption is that $\boldsymbol{L}$ is approximately Gaussian distributed (for suitable choice of the estimator $\hat{\lambda}$) with mean $\overline{\boldsymbol{L}}$ and covariance matrix $\boldsymbol{\Sigma}$. The density of $\boldsymbol{L}$ is given by: $f_L(\boldsymbol{l}) = \mathcal{N}(\boldsymbol{l}; \overline{\boldsymbol{L}}, \boldsymbol{\Sigma}) =$

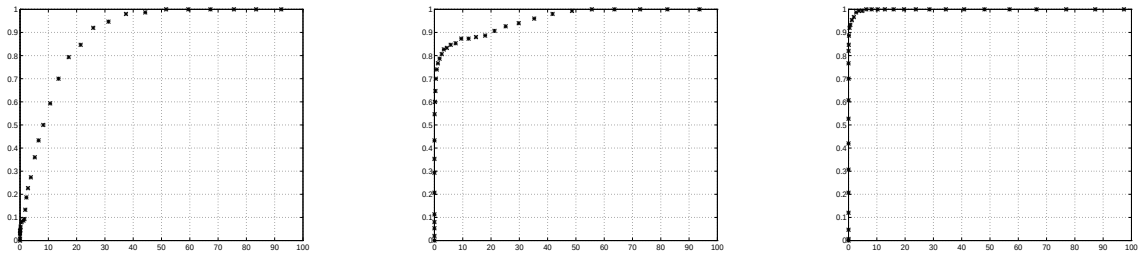
$$\frac{1}{(2\pi)^{\frac{n}{2}} |\det(\boldsymbol{\Sigma})|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}(\mathbf{l} - \overline{\mathbf{L}})^{\mathbf{T}} \boldsymbol{\Sigma}^{-\mathbf{1}} (\mathbf{l} - \overline{\mathbf{L}})\right) \quad (3)$$

where n is the number of components in L. (For N features, n is $N(N-1)/2$. The statistics $\overline{\boldsymbol{L}}$ and $\boldsymbol{\Sigma}$ can be estimated from training data.

Estimation of Scale: The apparent size of a face varies significantly depending on the proximity of the subject to the camera. For example, if the camera is twice as close, the mutual distances between the facial features will be twice as large. Scale invariance can be achieved by estimating the scale and then dividing it out.

Given an incoming vector $\boldsymbol{X}$ of natural mutual distances, we use the maximum likelihood (ML) approach to estimate the scale. Conditioned upon λ, the density of $\boldsymbol{X}$ is given by $f_X(\boldsymbol{x}|\lambda) = \mathcal{N}(\boldsymbol{x}; \lambda\overline{\boldsymbol{L}}, \lambda^2\boldsymbol{\Sigma}) =$

$$\frac{1}{(2\pi)^{\frac{n}{2}} |\det(\boldsymbol{\Sigma})|^{\frac{1}{2}} \lambda^{\mathrm{n}}} \cdot \exp\left(-\frac{1}{2}(\frac{\mathbf{x}}{\lambda} - \overline{\mathbf{L}})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\frac{\mathbf{x}}{\lambda} - \overline{\mathbf{L}})\right) \quad (4)$$



Right-Eye *Nose/Lip* *Left-Nostril*

Figure 1: Probability of detecting the true feature vs. the average number of *false-alarms* per image.

The ML estimate for the scale is the value $\hat{\lambda}$ which solves $\frac{\partial}{\partial \lambda} \log f_X(\boldsymbol{x}|\lambda) = 0$, i.e.,

$$\hat{\lambda} = \left(\frac{\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}\right) \cdot \frac{2}{1 + \sqrt{1 + \frac{4n(\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x})}{(\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu})^2}}} \quad (5)$$

If we have a prior density over λ, we could instead use the MAP (*maximum a posteriori*) estimate. Notice however that if λ is distributed uniformly over a range of interest, the MAP estimate is the same as the ML estimate, with the exception that estimates outside the range are clipped.

3.2 Controlled Search

As discussed at the beginning of Section 3, the brute-force approach for graph matching (even in the case of labeled graphs) is computationally very demanding. In this section, we show how the complexity can be greatly reduced by exploiting the statistical structure of the graph. The basic idea is that given the positions of several features, we can estimate the locations of the other features *and the covariance of the estimates.* In the brute-force method the original features would have to be coupled with all combinations of candidates for the other features. However, by knowing where the other features should be found and how much variance exists in their expected locations, we can limit the number of constellations formed. The original features will only be coupled with candidates for the other features that appear "close" to the expected locations, where the meaning of "close" is determined by the covariance estimates.

To begin with, assume we have a vector of scale-normalized mutual distances $\boldsymbol{L}$ in which some of the distances are observed and others are missing. Denote the *observed* data by $\boldsymbol{O}$ and the *missing* data by $\boldsymbol{M}$. Given the *observed* data $\boldsymbol{O} = \boldsymbol{o}$, there are a number of estimators that could be used to estimate the *missing* data. A particularly useful choice is the conditional mean since this estimator has the minimum variance property [2]. For jointly Gaussian distributed data, the conditional mean estimator is given by $\hat{\boldsymbol{M}}(\boldsymbol{O}) = \mathrm{E}(\boldsymbol{M}|\boldsymbol{O}) =$

$$\overline{\boldsymbol{M}} + \boldsymbol{\Sigma}_{mo} \boldsymbol{\Sigma}_{oo}^{-1} (\boldsymbol{O} - \overline{\boldsymbol{O}}) \quad (6)$$

Hence, given the observation $\boldsymbol{O} = \boldsymbol{o}$, we would estimate $\hat{\boldsymbol{m}} = \hat{\boldsymbol{M}}(\boldsymbol{o})$. Further, the conditional error covariance is $\boldsymbol{\Sigma}_{\hat{m}\hat{m}} = \mathrm{E}((\boldsymbol{M} - \hat{\boldsymbol{m}})(\boldsymbol{M} - \hat{\boldsymbol{m}})^T | \boldsymbol{O} = \boldsymbol{o}) =$

$$\boldsymbol{\Sigma}_{mm} - \boldsymbol{\Sigma}_{mo} \boldsymbol{\Sigma}_{oo}^{-1} \boldsymbol{\Sigma}_{om} \quad (7)$$

If instead of working with the scale-normalized mutual distances, we choose to work with the natural mutual distances, we must take into account the estimated scale parameter $\hat{\lambda}$. The natural mutual distances are useful because these can be used to directly specify where to look for points *in the image.* If we assume that we observe only a partial vector of natural mutual distances $\boldsymbol{x}_o$, the best estimate for the missing data $\hat{\boldsymbol{x}}_m$ is simply

$$\hat{\boldsymbol{x}}_m = \hat{\lambda}(\overline{\boldsymbol{M}} + \boldsymbol{\Sigma}_{mo} \boldsymbol{\Sigma}_{oo}^{-1} (\frac{\boldsymbol{x}_o}{\hat{\lambda}} - \overline{\boldsymbol{O}})) \quad (8)$$

$$\boldsymbol{\Sigma}_{\hat{x}_m \hat{x}_m} = \hat{\lambda}^2 (\boldsymbol{\Sigma}_{mm} - \boldsymbol{\Sigma}_{mo} \boldsymbol{\Sigma}_{oo}^{-1} \boldsymbol{\Sigma}_{om}) \quad (9)$$

where the scale estimate $\hat{\lambda}$ is also based on the observed data.

So far we have shown that given a partial constellation (some features missing), we can estimate the missing distances using Equation 6 and also compute the error covariance of our estimate. To translate this information back to the image plane, suppose that we are given two points and the distances of a third point from these two. Then, the third point can be located unambiguously up to a mirror reflection. Let d_{13} and d_{23} be the distances from the two points, which we denote by $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The position of the third point $\boldsymbol{x}_3$ is given by:

$$\mathbf{x_3} = \frac{d_{13}}{d_{12}} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} (\mathbf{x_2} - \mathbf{x_1}) + \mathbf{x_1} \quad (10)$$

where

$$\cos\theta = \frac{d_{13}^2 + d_{12}^2 - d_{23}^2}{2 d_{13} d_{12}} \quad (11)$$

and the possibility of mirror reflection is due to the sign ambiguity of $\sin\theta$. The conditional error covariance $\boldsymbol{\Sigma}_{\hat{x}_m \hat{x}_m}$ can give us the confidence of the calculated position of the third point. Let $\boldsymbol{d} = [d_{13}\ d_{23}]^T$

and $\boldsymbol{x}_3 = \boldsymbol{F}(\boldsymbol{d})$, where $\boldsymbol{F}$ is the function defined by Equations 10 and 11. This function can be linearized using the Taylor series:

$$\boldsymbol{x}_3 \approx \boldsymbol{F}(\overline{\boldsymbol{d}}) + \boldsymbol{DF}(\overline{\boldsymbol{d}})(\boldsymbol{d} - \overline{\boldsymbol{d}}) \quad (12)$$

$$\boldsymbol{DF} = \begin{bmatrix} \frac{\partial \boldsymbol{F}_1}{\partial d_{13}} & \frac{\partial \boldsymbol{F}_1}{\partial d_{23}} \\ \frac{\partial \boldsymbol{F}_2}{\partial d_{13}} & \frac{\partial \boldsymbol{F}_2}{\partial d_{23}} \end{bmatrix} \quad (13)$$

With this linearization, we obtain

$$\boldsymbol{\Sigma}_{\boldsymbol{x}_3} = (\boldsymbol{DF})\boldsymbol{\Sigma}_d(\boldsymbol{DF})^T \quad (14)$$

The covariance matrix for $\boldsymbol{x}_3$ specifies an *ellipse* where we should look for the missing features.

Examples of how this estimation procedure works are shown in Figure 2. We have five features in these examples: the two eyes, the nose/lip junction, and the two nostrils. Given two of these features, we estimate the position of the remaining features and use the error covariance to specify ellipses which will contain the true feature with probability 0.9999.

Figure 2: The locations of the missing features were estimated from two points. The ellipses show the areas which with high probability include the missing features.

The controlled search idea is implemented as follows. Two candidate features are selected from the incoming image. These features must be *strong features*, meaning that their quality of match (Equation 2) must exceed a higher threshold τ_{hi}. The expected locations and error ellipses for the other features are then estimated as described above. Constellations are formed only from candidates that lie inside the appropriate search ellipses. This process is repeated for each pair of candidate features.

Prior knowledge can also be used to limit the search. For example, suppose it is known a priori that the scale of the face should be in a particular range or that the faces will be upright. This information could be used to immediately reject some feature pairs, e.g., don't form any constellations from two eye candidates that are too far apart.

Complexity Analysis: Let M_1 be the average number of strong candidates for each feature, M_2 the average number of regular candidates, and K the average number of candidates inside each search ellipse. The basic premise of the controlled search is that in most cases K will be much less than M_2. The typical number of constellations considered by the controlled search algorithm is $O(N^2 M_1^2 K^{N-2})$, where N is the number of facial features used. For comparison, the brute-force method (also using two-tiered thresholding) would search through $O(N^2 M_1^2 M_2^{N-2})$ constellations. On average the controlled search reduces the number of constellations to be considered by the factor $(\frac{M_2}{K})^{N-2}$.

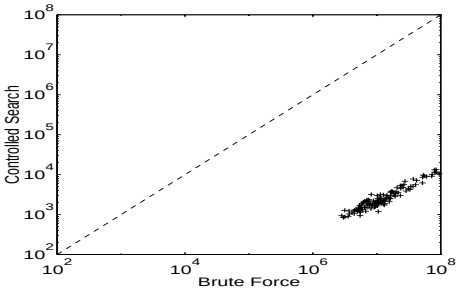


Figure 3: The number of constellations evaluated by controlled search vs. the brute-force approach.

The effectiveness of the controlled search method in practice is illustrated in Figure 3 for the experiments of Section 4. For each image, the number of constellations produced by the controlled search is plotted against the number of constellations that would have been produced by the brute-force method. The average number of constellations per image was approximately 3.5×10^3 for controlled search and 2.2×10^7 for brute-force. Hence, the reduction in computational load was about 4 orders of magnitude. Also, notice that in the worst case, number of constellations generated by the controlled search method was on the order of 10^4.

3.3 Ranking of Constellations

Given two point constellations $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, how do we decide which one is more face-like? Since we want invariance with respect to translation, rotation, and scale, we should transform to vectors of scaled mutual distances. The problem can now be rephrased as a test of hypothesis for the joint observation $[d(\boldsymbol{z}_1);\ d(\boldsymbol{z}_2)]$. The first hypothesis H_1 is that $\mathbf{z}_1$ is from a face and $\mathbf{z}_2$ is not. The competing hypothesis H_2 is that $\mathbf{z}_2$ is from a face and $\mathbf{z}_1$ is not. Denoting the probability density of the vector of mutual distances conditioned on being a face by $p(d(\boldsymbol{z})|F)$ and conditioned on not being a face by $p(d(\boldsymbol{z})|\overline{F})$, standard results from decision theory [8, 9] show that the optimal discriminant is

$$L_* = \frac{p(d(\boldsymbol{z}_1)|F) \cdot p(d(\boldsymbol{z}_2)|\overline{F})}{p(d(\mathbf{z}_1)|\overline{F}) \cdot p(d(\boldsymbol{z}_2)|F)} \quad (15)$$

which can be rewritten as

$$L_* = \frac{L(d(\boldsymbol{z}_1))}{L(d(\boldsymbol{z}_2))} \quad (16)$$

where

$$L(d(\boldsymbol{z})) \triangleq \frac{p(d(\boldsymbol{z})|F)}{p(d(\boldsymbol{z})|\overline{F})} \quad (17)$$

Equation 17 provides the proper function for ranking a constellation $\mathbf{z}$ according to how face-like it is. In words, the ranking function is just the probability that $\mathbf{z}$ corresponds to a face versus the probability it was generated by an alternative mechanism (to be discussed further below). The constellation receiving the highest ranking value L will defeat any other constellation in a head-to-head comparison to decide which is more face-like.

Incomplete constellations may be ranked using the following method: $p(d(\mathbf{z})|F)$ is the marginal probability over the *observed* features multiplied by the probability of encountering the observed features conditioned on being a face:

$$p(\mathbf{o}|F) = \prod_{\mathbf{o}} \gamma_i \prod_{\overline{\mathbf{o}}} (1-\gamma_i) \quad (18)$$

where γ_i is the probability that the true location of the i^{th} feature is detected. Here we are making an assumption that the detectors fail independently. This assumption is probably reasonable provided the face remains quasi-frontal. Clearly, however, when the face is significantly rotated in depth the features on one side of the face will be more likely to both be missed.

Now, we have to define the probability density for the alternative hypothesis, $\overline{F}$. This is complicated by the possibility that some candidate constellations may consist of $n < N$ true features and $N-n$ bad features. We can expand $p(d(\mathbf{z})|\overline{F})$ as:

$$\frac{\sum p(d(\mathbf{z})|b_1,\ldots,b_N) \cdot Pr(b_1,\ldots,b_N)}{\sum Pr(b_1,\ldots,b_N)} \quad (19)$$

where $b_i = 0$ or 1 depending on whether feature i is a false alarm or the true feature. The summations above go over all N-tuples having at least one $b_i = 0$.

The terms $Pr(b_1, b_2, \ldots, b_N)$ are the prior probabilities of observing a constellation in which the i^{th} feature is of type b_i, where $b_i = 1$ for true features and $b_i = 0$ for false alarms. Assuming independence, we can write $Pr(b_1,\ldots,b_N) = \prod_{i=1}^{N} Pr(b_i)$ with

$$Pr(b_i = 0) = (1-\gamma_i) + \gamma_i \cdot \frac{\overline{m_i} - 1}{\overline{m_i}} \quad (20)$$

$$Pr(b_i = 1) = \frac{\gamma_i}{\overline{m_i}} \quad (21)$$

Here, $\overline{m_i}$ is the average number of candidates located for the i^{th} feature. These equations tell us that the prior probability that a certain feature is the true one ($Pr(b_i = 1)$) is the probability it is successfully detected and then chosen from the candidates in an image. On the other hand, the prior probability that a feature is a false alarm ($Pr(b_i = 0)$) is the probability the true feature is missed or the true feature is detected but not selected from the candidate pool.

Now, we consider the terms $p(d(z)|b_1,\ldots,b_N)$. Allowing for the possibility that z is an incomplete constellation, $p(d(z)|b_1,\ldots,b_N)$ is:

$$p(d(z)|o, b_1,\ldots,b_N) \cdot p(o|b_1,\ldots,b_N) \quad (22)$$

Here, the probability of observing a certain, possibly incomplete, constellation given the b_i's is

$$p(o|b_1,\ldots,b_N) = \prod_{o} Pr(\mathbf{h_i}|b_i) \prod_{\overline{o}} Pr(\overline{\mathbf{h_i}}|b_i) \quad (23)$$

where $\mathbf{h_i}$ means that the i^{th} feature in the constellation is observed and $\overline{\mathbf{h_i}}$ the i^{th} feature is missing. Simple reasoning leads to the following values:

$$Pr(\mathbf{h_i}|b_i = 1) = \gamma_i \quad (24)$$

$$Pr(\overline{\mathbf{h_i}}|b_i = 1) = 1 - \gamma_i \quad (25)$$

$$Pr(\mathbf{h_i}|b_i = 0) = \frac{\overline{m}_i - 1}{\overline{m}_i} \quad (26)$$

$$Pr(\overline{\mathbf{h_i}}|b_i = 0) = \frac{1}{\overline{m}_i} \quad (27)$$

4 Experimental Results

4.1 Face Databases

We tested our algorithm on two different databases of images: (1) the Lab Sequence and (2) the Studio Database. The Lab Sequence is a very challenging database collected under realistic circumstances at an ordinary computer laboratory. The subjects were seated 2–3 meters away from the camera and allowed to move freely and make different facial expressions. The background was complicated and continually changing due to people walking around behind the subjects. The entire Lab Sequence contains 900 images, of which we have currently tested on the first 150. Two of these are shown in Figure 4.

Figure 4: Typical images from the Lab Sequence. These examples show changes in scale and orientation, as well as occlusions. Also notice that the background is not constant.

The Studio Database contains images taken under well-controlled conditions. There are 180 images of

18 subjects. The subjects were imaged at a distance of two meters against a plain white background. All views were quasi-frontal and were collected under the same lighting conditions. Typical images are shown in Figure 5. This database was used primarily to train the algorithm and to evaluate performance in a benign environment.

Figure 5: Typical images from the Studio Database.

4.2 Parameters

Detectors were synthesized for five features on the face: the two eyes (LE & RE), the two nostrils (LN & RN), and the nose/lip junction (NL). The template response vectors were obtained by averaging the response vectors over three faces in the same dataset. The performance of the detectors on the Lab Sequence (for three of the detectors) was shown previously in Figure 1. The detection thresholds τ_{hi} and τ_{th} used in our experiments along with the resulting parameters γ_i and $\bar{m}_i$ are shown below:

	LE	RE	NL	LN	RN
τ_{hi}	0.63	0.72	0.74	0.85	0.52
τ_{th}	0.51	0.65	0.66	0.75	0.50
γ_i	0.70	0.70	0.85	0.92	0.78
$\bar{m}_i$	16	14	8	2	9

The parameters $\overline{\boldsymbol{L}}$ and $\boldsymbol{\Sigma}$, which define the mean and covariance for the vectors of scaled mutual distances, were estimated from 112 faces in the Studio Database; the estimated values are tabulated in Table 1. These parameters were used for all the experiments reported below since the distribution of the mutual distances between facial features is expected to be the same across databases.

The statistics for the $\overline{F}$ distribution in Equation 17 were estimated using a Monte-Carlo simulation. We generated random points in the image and estimated the mean and covariance for the mutual distances. We divided the mutual distances into two groups: *F-Edges and* $\overline{F}$-Edges. If an edge's two end-points are believed to be true features (i.e., if in Equation 19, $b_i = 1$ for both end-points), it is called an *F-Edge*; otherwise, it is called an $\overline{F}$-*Edge*. The mean for all *F-Edges* and covariance between two *F-Edges* are taken from Table 1, while all other parameters are obtained from the Monte-Carlo simulation.

4.3 Performance

Before reporting the performance of the algorithm, we must define some terminology. Since we allow missing nodes in a graph, incomplete constellations that correctly match a subgraph of the face are considered *equivalent* to matching the full graph. Therefore, any constellations which correctly locate three or more features on the face are considered to be correct. (An eye feature is considered correctly located if it is detected within 9 pixels of the true position. For the nose features, the threshold is 5 pixels.) Of course, constellations with one or more incorrect features are considered to be incorrect. The highest-ranked correct constellation will be referred to as the *best correct match* while the highest-ranked incorrect constellation will be referred to as the *best incorrect match*.

The face localization algorithm was applied to the first 150 frames of the Lab Sequence. On this database, we were able to achieve a correct localization rate of 86%. Incorporating a constraint that the estimated head orientation be within 45° of upright increased the performance to 89%. These results are somewhat pessimistic because the algorithm is only designed for quasi-frontal images, but approximately 7% of the images in the Lab Sequence show rotations in depth $> 15°$. After retabulating the results over just the quasi-frontal views, we found the performance to be 95%.

The performance on three typical images is shown in Figure 7. The first column shows the highest-ranked constellation (a correct match in each case) and the second column shows the highest-ranked incorrect constellation. The third column shows the ranking scores for the top 30 constellations, where a "+" denotes a correct match and an "o" denotes an incorrect match. The two horizontal dotted lines appearing in each plot highlight the difference between the ranking score of the best correct match and the best incorrect match.

The ability of the algorithm to easily discriminate between correct and incorrect matches provides a measure of the system robustness. Figure 6 shows the cumulative percentage of images for which the best correct match exceeds the best incorrect match by a given order of magnitude. For example, the solid horizontal line shows that in 80% of the images the best correct match is one order of magnitude better than the best incorrect match.

One deficiency in our experimentation is that we have analyzed only the Lab Sequence with our localization system, and all these images show the same individual. However, using a precursor to our current algorithm (different feature detectors and a different ranking function), we were able to achieve about 80% performance on the Studio Database, which contains 180 images of 18 individuals. Therefore, we believe the current system will generalize well to other individuals.

5 Conclusions

We have developed an algorithm to locate quasi-frontal faces in cluttered scenes. The algorithm consists of three steps: using local detectors to identify candidate feature points, forming constellations from within the pool of candidate features, and ranking the constellations based on a probability density. The algorithm is translation, rotation (provided the face re-

	$\overline{L}$	Σ									
RE-LE	28.79	4.50	0.06	-0.27	0.20	0.25	-0.44	0.49	-0.42	-0.27	-0.59
RE-NL	23.97	0.06	2.19	1.46	1.78	1.33	0.58	2.03	1.54	0.13	0.67
LE-NL	23.57	-0.27	1.46	3.29	1.39	2.90	0.27	1.31	2.92	0.69	0.81
RE-RN	19.29	0.20	1.78	1.39	1.96	1.21	-0.06	1.78	1.59	0.08	0.13
LE-RN	24.65	0.25	1.33	2.90	1.21	2.88	0.37	1.18	2.58	0.53	0.85
NL-RN	5.17	-0.44	0.58	0.27	-0.06	0.37	0.77	0.35	0.16	0.07	0.71
RE-LN	25.95	0.49	2.03	1.31	1.78	1.18	0.35	2.19	1.35	0.32	0.70
LE-LN	20.02	-0.42	1.54	2.92	1.59	2.58	0.16	1.35	3.16	0.12	0.33
NL-LN	4.65	-0.27	0.13	0.69	0.08	0.53	0.07	0.32	0.12	0.76	0.68
RN-LN	8.95	-0.59	0.67	0.81	0.13	0.85	0.71	0.70	0.33	0.68	1.21

Table 1: The sample mean and covariances estimated from 112 faces in the Studio Database.

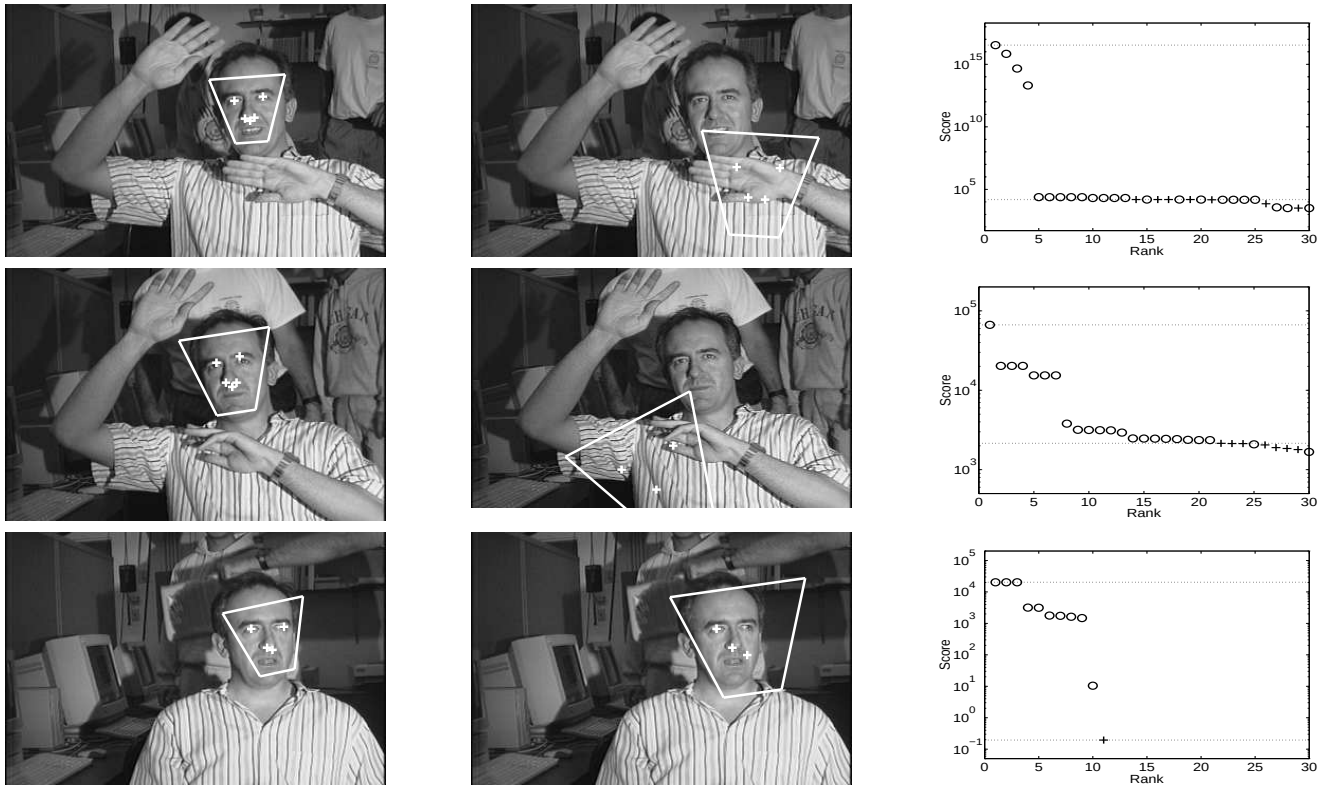


Figure 7: Typical performance on three images from the Lab Sequence. The first column shows the best correct match, while the second column shows the best incorrect match. The third column shows the ranking score for the top 30 matches.

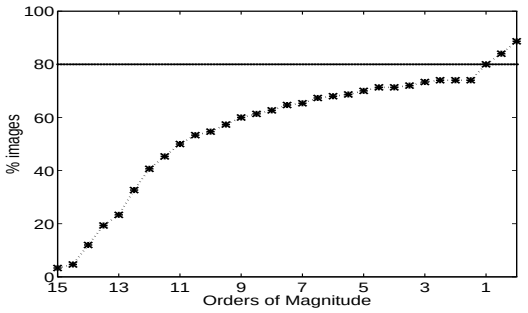


Figure 6: Cumulative percentage of images versus the difference in score between the best correct match and the best incorrect match. The horizontal line indicates that in over 80% of the images, the difference is at least one order of magnitude. The rightmost point shows that the overall localization rate is close to 90%.

mains quasi-frontal), and scale invariant and can handle occlusions. Although the algorithm uses graph matching, the probability density over the graph is exploited to contain the computational complexity at a reasonable level. Performance of the algorithm was 95% over quasi-frontal views of faces in a *realistic* sequence of images.

Several extensions to the algorithm are underway. Each feature detector currently uses a single template response vector obtained from three individuals in the Studio Database. Using more training data should improve the detector performance considerably. However, here we want to emphasize the following point: we believe the exact form of the feature detectors is not critical. Another scheme such as matched filtering or eigen-features could be used without ruining the overall performance. Also, no matter what scheme is used for feature detection, we do not expect that the true features can be located with 100% probability and zero false alarms. Thus, the spatial arrangement of the feature points must be considered in order to produce a robust face localization algorithm.

A second extension involves the representation of constellations as a vector of mutual distances. This representation causes some difficulties in transforming back to the image domain and some theoretical difficulties with the probability distributions. However, we believe we can fix these problems by using the theory of *shape statistics* [7, 5].

Finally, our statistical model of the random graphs currently only applies to quasi-frontal faces. We envision two ways of extending our method to full rotations in depth. The first method is to use separate models for frontal, three-quarters, and profile. The feature vector matching would also need to be modified since some features look quite different depending on the viewing angle. The other method we are considering is to model the variations in feature positions in 3-D and then look at how these variations project into 2-D.

Acknowledgement

This work was supported by the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation (NSF) Engineering Research Center Program, and by the California Trade and Commerce Agency, Office of Strategic Technology. Additional funding was provided by ONR grant n.N00014-93-1-0990, an NSF National Young Investigator award, a grant from Intel, and a grant from Caltech. We are also very grateful to Jitendra Malik for useful discussions.

References

[1] Y. Amit and A. Kong. "Graphical Templates for Image Matching". Technical Report 373, Department of Statistics, University of Chicago, August 1993.

[2] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, 1979.

[3] R. Brunelli and T. Poggio. "Face Recognition: Features versus Templates". *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(10):1042–1052, October 1993.

[4] G. Burel and D. Carel. "Detection and Localization of Faces on Digital Images". *Pattern Recognition Letters*, pages 963–967, Oct 1994.

[5] M.C. Burl, T.K. Leung, and P. Perona. "Face Localization via Shape Statistics". In *Proc. International Workshop on Face and Gesture Recognition*, Zurich, Switzerland, June 1995.

[6] P. Burt. Private Communication, 1995.

[7] I.L. Dryden and K.V. Mardia. "General Shape Distributions in a Plane". *Adv. Appl. Prob.*, 23:259–276, 1991.

[8] R.O. Duda and P.E. Hart. *"Pattern Classification and Scene Analysis"*. Wiley, 1973.

[9] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[10] D. Jones and J. Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Proc. 2^{nd} Europ. Conf. Comput. Vision, G. Sandini (Ed.), LNCS-Series Vol. 588, Springer-Verlag*, pages 395–410, 1992.

[11] M.S. Kamel, H.C. Shen, A.K.C. Wong, and T.M. Hong et al. "Face Recognition using Perspective Invariant Features". *Pattern Recognition Letters*, 15(9):877–883, Sept 1994.

[12] T. Kanade. "Computer Recognition of Human Faces". *Interdisciplinary Systems Research*, 47, 1977.

[13] H-Y.S. Li, Y. Qiao, and D. Psaltis. "Optical Network for Real-time Face Recognition". *Applied Optics*, 32(26):5026–5035, Sept 1993.

[14] A.J. O'Toole, H. Abdi, K.A. Deffenbacher, and D.Valentin. Low-dimensional representation of faces in higher dimensions of the face space. *J. Opt. Soc. Am. A*, 10(3), 1993.

[15] H.L. Van Trees. *Detection, Estimation, and Modulation Theory: Part 1*. John Wiley and Sons, 1968.

[16] M. Turk and A. Pentland. "Eigenfaces for Recognition". *J. of Cognitive Neurosci.*, 3(1), 1991.

[17] D. Valentin, H. Abdi, A.J. O'Toole, and G.W. Cottrell. "Connectionist Models of Face Processing: A Survey". *Pattern Recognition*, 27(9):1209–30, Sept 1994.

[18] G. Yang and T.S. Huang. "Human Face Detection in a Complex Background". *Pattern Recognition*, 27(1):53–63, Jan 1994.

[19] A.L. Yuille. "Deformable Templates for Face Recognition". *J. of Cognitive Neurosci.*, 3(1):59–70, 1991.