Coarse-to-Fine Face Detection

François Fleuret [*]   Donald Geman [†]

June 2000

[*] Avant-Projet IMEDIA, INRIA-Rocquencourt, Domaine de Voluceau, B.P.105, 78153 Le Chesnay. Email:Francois.Fleuret@inria.fr. Supported in part by the CNET.

[†] Department of Mathematics and Statistics, University of Massachusetts, Amherst, MA 01003. Email:geman@math.umass.edu. Supported in part by ONR under contract N00014-97-1-0249 and ARO under MURI grant DAAH04-96-1-0445.

Abstract

We study visual selection: Detect and roughly localize all instances of a generic object class, such as a face, in a greyscale scene, measuring performance in terms of computation and false alarms. Our approach is sequential testing which is coarse-to-fine in both in the exploration of poses and the representation of objects. All the tests are all binary and indicate the presence or absence of loose spatial arrangements of oriented edge fragments. Starting from training examples, we recursively find larger and larger arrangements which are "decomposable," which implies the probability of an arrangement appearing on an object decays slowly with its size. Detection means finding a sufficient number of arrangements of each size along a decreasing sequence of pose cells. At the beginning, the tests are simple and universal, accommodating many poses simultaneously, but the false alarm rate is relatively high. Eventually, the tests are more discriminating, but also more complex and dedicated to specific poses. As a result, the spatial distribution of processing is highly skewed and detection is rapid, but at the expense of (isolated) false alarms which, presumably, could be eliminated with localized, more intensive, processing.

1 Introduction

We study face detection in the framework of learning-based, visual selection: Starting with a training set of examples of a generic object class, in our case a "face," detect and roughly localize all instances of this class in greyscale scenes. The training examples are subimages containing a single instance of the object at various poses, for example frontal views of faces at a range of scales, tilts, etc. Whereas the backgrounds in the training samples might be very simple, the detection algorithm must function in natural, highly cluttered scenes.

Performance is measured by the false alarm rate and the amount of (on-line) computation necessary to achieve a very small false negative rate, albeit with an imprecise determination of the pose. In fact, we are going to emphasize computation; presumably, sufficiently isolated false alarms could be removed, and better localization achieved, with more intensive but highly localized processing, and therefore with a modest increase in computation. Finally, other performance factors might also be important, such as memory, the size of the training set, and the duration of training.

The problem of detecting instances from a generic object class has of course been studied in the computer vision literature. We restrict our attention to detecting (but not recognizing) faces, and without information due to color, depth or motion. The generality of our approach is discussed in the concluding section; any potential limitations should then be apparent.

A variety of methods have been proposed for face detection, including artificial neural networks (Rowley, Baluja & Kanade 1998), (Sung & Poggio 1998), support vector machines (Osuna, Freund & Girosi 1997), graph-matching (Leung, Burl & Perona 1995), (Maurer & von der Malsburg 1996), Bayesian inference (Cootes & Taylor 1996), deformable templates (Miao, Yin, Wang, Shen & Chen 1999),(Yuille, Cohen & Hallinan 1992) and those based on color (Haiyuan, Qian & Masahiko 1999),(Sabert & Tekalp 1998) and motion (Ming & Akatsuka 1998), (Wee, Ji, Yoon & Park 1998). The precursor of this work is (Amit & Geman 1999): Features are spatial arrangements of edge fragments, induced from training faces at a reference pose, and computation is minimized via a generalized Hough transform; there is no on-line optimization and no segmentation apart from visual selection itself. In evaluating our results, we are also going to focus on comparisons with the work in (Rowley 1999) and (Rowley et al. 1998) since this seems to be among the most comprehensive studies as well as a fair representation of the state-of-the-art.

This work stems from a broader project on visual recognition as a "twenty questions game," in other words a problem in efficient sequential testing. This theme was pursued in the context of classification trees and stepwise entropy reduction in (Amit & Geman 1997), (Geman & Jedynak 1996), (Jedynak & Fleuret 1996) and

(Wilder 1998). The detection counterpart of classification is sequential testing in order to discover which of two classes is true; one is the target and the other, "background," is dominant. For example, we seek to identify one famous person from among all others, a compound alternative which is a priori much more likely. The target is represented as a conjunction of elementary attributes (for instance, Napoleon is simultaneously *deceased, general, Corsican, etc.*) which can be checked in any order.

If the "cost" of checking every attribute is the same, then naturally a good procedure is to check them in their order of likelihood relative to the dominant class - from rare ones to common ones. In this way the search is over quickly *on the average*, but never fails to detect the target. However, if there are numerous target variations and if common attributes (relative to the background population) appear in many representations, then it makes sense to make "testing" for common attributes relatively cheaper than for rare ones, in which case it may be more globally efficient to proceed instead from common to rare. This is the case, for instance, if the cost of testing an attribute is its negative log-likelihood (as in coding). This type of reasoning motivates our sequential testing strategy: The backbone of the detection algorithm is a "coarse-to-fine" tree structure which minimizes average computation under a certain statistical model for cost and likelihood.

In visual processing, the corresponding attributes are binary image functionals; in fact, throughout this paper, all features are binary, and referred to as "tests." The object class is no longer a simple conjunction, but rather, like the background class, an enormous *disjunction of conjunctions.* The individual conjunctions correspond to distinguished object features when the pose and lighting are known to very high precision. The disjunctions account for general poses (locations, scales, orientations) as well as finer variations due to lighting and local, nonlinear shape deformations. Of course efficient detection implies a high degree of invariance - capturing these disjunctions succinctly, without explicit enumeration.

The most elementary tests correspond to local edge fragments. The fragments have an approximate location and an approximate orientation; the definition is purposely loose in order to accommodate geometric invariance. The other tests are products (conjunctions) of elementary ones, and hence correspond to the presence or absence of a spatial arrangement of edge fragments. They have no a priori semantical interpretation; the construction is purely statistical and learning-based. The key property of the products is "decomposability": each product can be divided into two correlated subproducts, each of which further splits into two correlated smaller subproducts, and so forth all the way down to the elementary tests. The motivation is that the probability that a decomposable test of size k appears on an object instance decreases gradually as k increases compared with the decrease in general backgrounds - in fact exponentially with $\log_2 k$ instead of k (§6).

The testing strategy is based on a sequence of nested partitions of the set of possible poses. The strategy is coarse-to-fine in the generality of the pose, and coarse-to-fine in complexity at each level of generality. In order to declare detections, we successively visit cells in these partitions and successively check for a minimal number of decomposable tests of each complexity. The order of visitation is adaptive and chosen to minimize overall computation. Initially, the conjunctions are simple and sparse (e.g., involve only a few non-localized, non-specific edge fragments), and thereby accommodate many poses simultaneously; eventually they are more dense (i.e., larger numbers of more specialized fragments), and hence more dedicated to specific poses. The result is that flat areas and other "non-object-like" portions of the image are rejected very quickly and with very simple tests. Highly cluttered areas require more processing and faces the most of all. In Figure 1 we show an illustration of the spatial distribution of processing corresponding to the scene in Figure 2; it is very highly concentrated in the area of detections.

The experiments involve scenes with frontal views of faces. We train with a portion of the Olivetti database - 300 faces representing 10 pictures of each of 30 individuals.

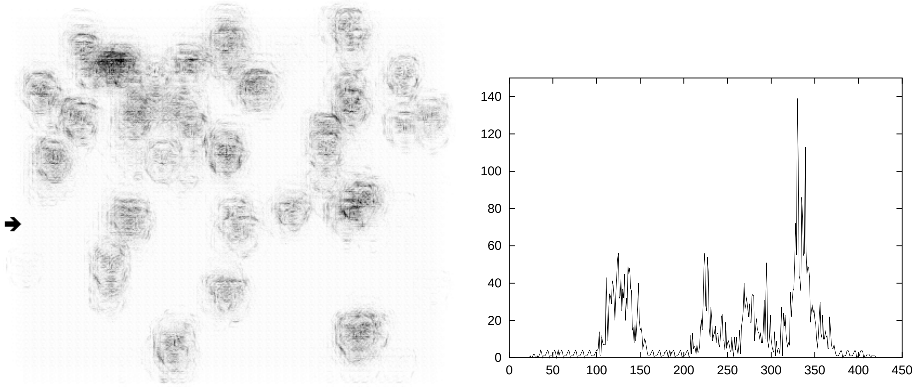


Figure 1: The coarse-to-fine nature of the algorithm is illustrated by counting, for each pixel, the number of times the detector checks for the presence of an edge in its vicinity. Left: The grey level is proportional to this count. Right: The scan line corresponding to the arrow; it covers three faces.

The learning algorithm is a procedure for building larger and larger decomposable tests in a recursive, bottom-up fashion, and dedicated to specific pose cells. The algorithm for each cell is identical; only the training set changes. A relatively small training set is sufficient since we only use it to estimate correlations. In particular, we do not estimate a large system of coupled parameters as in other statistical learning methods.

One result is displayed in Figure 3. There are definitely false alarms, ranging from several to several tens depending on the scene, but the processing time and the number of missed faces are small relative to other algorithms; see §8. Hopefully, the confusions can be eliminated (without losing faces) with various ameliorations or with highly selective but relatively intensive processing, perhaps involving greyscale normalization and on-line optimization.

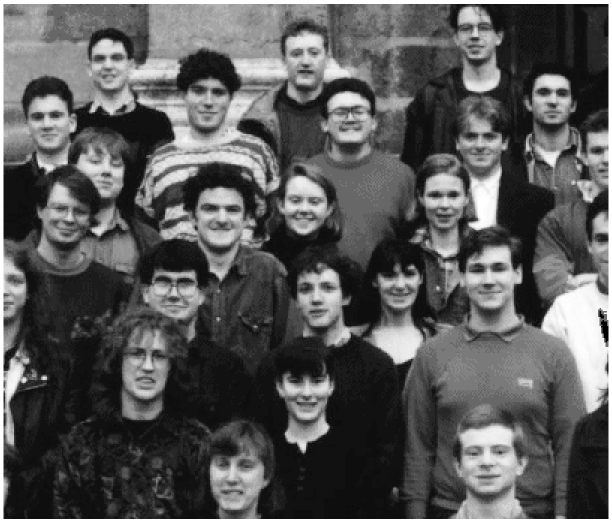


Figure 2: Example of a scene.

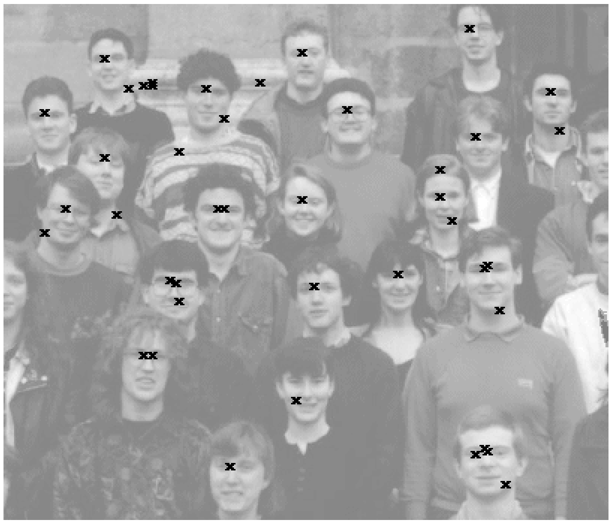


Figure 3: The detections in Figure 2.

2 Organization of the Paper

Since the algorithm is structured around nested partitions of "pose", we begin with that in §3. Given a "reference set" of poses, the mathematical set-up and performance criteria can made precise (§4). A summary of the detection and learning algorithms is given in §5; the constituents are then fleshed out in the remaining sections, except for a few technical arguments which appear in Appendices. Section 6 is devoted to the features we use, especially the notion of "decomposability" and a corresponding likelihood bound, and §7 explains how the decomposable arrangements - the main ingredients of the detector - are induced from training data. The sequential testing strategy for evaluating the detector is then described in §8 and experiments follow in §9. Finally, there is a critical evaluation of our approach in §10.

3 Pose Decomposition

The coarse-to-fine search is based on a hierarchical decomposition of the set of possible "poses" or "presentations" of an object. There is an invariant filter for each "cell" of the decomposition. In this paper the notion of pose is purely geometric, characterized by position, scale and orientation. However, even for a semi-rigid object such as a face, there are other aspects of an instantiation which carry valuable information for selection and discrimination, such as photometric parameters, more refined linear geometric properties and the existence of sub-components (e.g., glasses and beards). For some objects - including faces - it could be more efficient to recursively partition the presentations in a less dedicated way than is done here, thereby accommodating other important variations.

It is natural to define the pose of an object in terms of distinguished points. No corresponding features are defined; the points merely serve to define the pose. For faces, we use the positions of the eyes. Equivalently, the pose of a face has, by definition, a location (the midpoint between the eyes), a scale (the distance between the eyes) and a tilt (relative to the axis perpendicular to the segment joining the eyes). The position of the mouth is then roughly determined by the basic morphology of the face (although residual variations in the eye-to-mouth distance can be significant and could enter a finer decomposition). We do not attempt to detect frontal views of faces at all possible poses. Rather, the tilt (orientation) is restricted to $[-20^\circ, +20^\circ]$ and the scale to $10 - 160$ pixels. Consequently, we do not attempt to detect faces which are very tilted, very small or very large.

The invariant filters rely on common properties of faces over a range of poses. But faces at very different scales have very little shared structure, even if they are roughly superimposed. The same is true for two faces at approximately the same scale but far apart relative to that scale. Consequently, the coarsest pose cell we analyze invariantly accommodates all tilts but restricts the scale to the *reference range* of $10 - 20$ pixels and confines the location to the *reference block* of size 16×16. Let Θ denote this reference subset of poses. One can argue that the real detection problem does begin here; there is certainly enormous variability due to lighting, scale, tilt, local deformations, and of course different faces.

All the learning is dedicated to Θ. Faces in the scale range $20 - 160$ are detected by downsampling and rerunning the algorithm dedicated to Θ; faces at locations outside the reference block are detected by partitioning the image lattice into non-overlapping, 16×16 blocks. More details about these two "outer loops" are given in §5.

The set of poses Θ is partitioned M times by successive refinements. Let $\Lambda_{m,l}, l = 1, ..., L_m$, be the l'th cell of the m'th partition, $m = 0, 1, ..., M$. Here, $\Lambda_{0,1} = \Theta$ and for each $m = 1, ..., M$, the collection $\{\Lambda_{m,l}, l = 1, ..., L_m\}$ is a partition of Θ and a refinement of $\{\Lambda_{m-1,l}, l = 1, ..., L_{m-1}\}$. The complete family of cells is denoted by $\mathcal{C}$. In our experiments, $M = 5$. There are three quaternary splits on location ($16 \times 16 \rightarrow 8 \times 8 \rightarrow 4 \times 4 \rightarrow 2 \times 2$), and then one binary split on scale and one binary split on tilt. Modulo translation, this yields eleven different cells, as depicted

Location	Tilt	Scale
(in pixels)	(in degrees)	(in pixels)
16×16	-10 — 10	10 — 20
8×8	-10 — 10	10 — 20
4×4	-10 — 10	10 — 20
2×2	-10 — 10	10 — 20
2×2	-10 — 0	10 — 20
2×2	0 — 10	10 — 20
2×2	-10 — 0	10 — 14
2×2	-10 — 0	15 — 20
2×2	0 — 10	10 — 14
2×2	0 — 10	15 — 20

Table 1: Modulo translation, there are ten different "pose cells" in the hierarchy. Location, tilt and scale are defined in the text in terms of the positions of the two eyes. The finest cells are not very fine with respect to tilt and scale.

in Table 1. The finest cells localize the face within a 2×2 block and correspond to either "small scale" $(10 - 14)$ or "big scale" $(15 - 20)$, and to either "left tilt" $([-20^\circ, 0^\circ])$ or "right tilt" $([0^\circ, 20^\circ])$. Hence there are 256 fine cells. They are not really very "fine" but suffice to detect faces with a relatively small number of false alarms.

In Figure 4 we show a random sample of faces from the training set for each of three pose cells: The top group of faces have poses with location restricted to an 8×8 block, but no restrictions on tilt or scale; the middle group all have location in 2×2 block, right tilt, and scale in the full range $10 - 20$; and in the bottom group the same except that the scale is restricted to $15 - 20$.

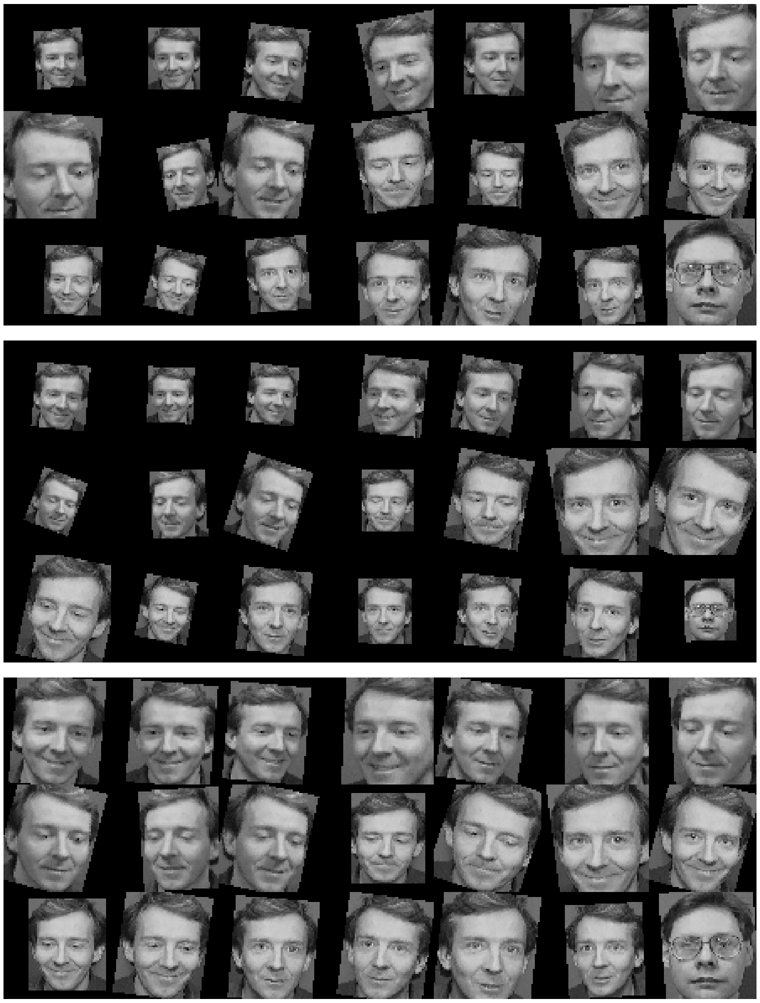


Figure 4: Random samples of training faces for each of three pose cells; they are synthetically generated from the original Olivetti database. Top: Location restricted to 8×8, all tilts and all (reference) scales; Middle: Location in 2×2, right tilts, all scales; Bottom: Location in 2×2, right tilts, large scales $(15 - 20)$.

4 Performance Constraints

As indicated earlier, the scenario we envision ("visual selection") is that the algorithm should be constructed to find *all* faces with very little computation, certainly well under one second for average-sized scenes. Weeding out the false positives is to be accomplished with more intensive but localized processing (or perhaps manually in some medical, military and other applications).

We can now be more precise about this formulation. Let $\mathcal{I}$ denote a set of (sub)images $I = \{I(u, v), (u, v) \in G\}$, say all "natural images," where G is a reference grid and $I(u, v)$ is quantized in a standard way, say to 256 grey levels. The images are partitioned into two subsets, "face" and "background," denoted $\mathcal{I}_F$ and $\mathcal{I}_B$. The face images contain a frontal view of a face with pose in Θ, where the corresponding 16×16 block is centered in G. All other images are background, even if there is a face at a pose outside Θ. Due to limiting the distance between the eyes to $10-20$ pixels, taking G of dimension 64×64 then accommodates all faces at reference poses.

Let P denote a probability measure on $\mathcal{I}$. We can think of P as the empirical measure on 64×64 subimages of all larger, natural images. Then P induces two conditional measures on $\mathcal{I}$: $P_0(.) = P(.|\mathcal{I}_B)$, the distribution on the background class, and $P_1(.) = P(.|\mathcal{I}_F)$, the distribution on the object class. Similarly, for any subset $\Lambda \subset \Theta$, we define P_Λ to be the induced probability measure on faces with a pose in Λ.

A *detector* is a mapping $f : \mathcal{I} \rightarrow \{0, 1\}$ where $f(I) = 0$ indicates "background" and $f(I) = 1$ indicates "face." The false negative error of f *relative to* Λ is $\alpha(f) = P_\Lambda(f = 0)$; the overall false negative error is $P_1(f = 0)$ and the false positive error is $P_0(f = 1)$. An *invariant detector* has $\alpha(f) = 0$.

In §8 we will define a random variable which is the cost of a procedure used to evaluate f. The mean cost with respect to P_0 represents the average amount of computation necessary to classify a background image. The motivation for the expectation relative to P_0 is that $P(\mathcal{I}_F) \ll P(\mathcal{I}_B)$; hence computational efficiency is driven by the rate at which background images are rejected as face candidates.

5 Summary of the Algorithm

There are really two algorithms - one for detection and one for learning. What follows is a summary of each one.

5.1 Detection

The detection algorithm has four nested loops. The two outer loops focus attention on a subset of scales and locations, namely a copy of Θ determined by a particular 64×64 subimage at a particular resolution. The two inner loops are the important ones and represent the coarse-to-fine search over refinements of the pose and over the complexity of the features. The outer loops are inherently parallel and the inner ones are serial.

One part of the outer loops is over resolutions. We downsample once (by averaging two-by-two blocks) in order to detect faces at scales $20 - 40$, twice to detect scales $40 - 80$, and thrice to detect scales $80 - 160$. The other part of the outer loop is over blocks. We partition the lattice into *non-overlapping* 16×16 blocks, and visit each one to determine if the image data in the surrounding 64×64 region supports the hypothesis of a face located there. Thus, at every resolution and in every block, we are only looking for faces at a reference pose. Surely there is some redundancy in separately analyzing the image data in each such region. For example, the basic local features are detected first throughout the image and other elements of the processing could be implemented more globally.

The two parts of the outer loop are depicted in Figure 5. The original image is on the left; it is downsampled once in the middle and twice on the right. In each case, the partition into non-overlapping 16×16 blocks is indicated by the overlaid grid.

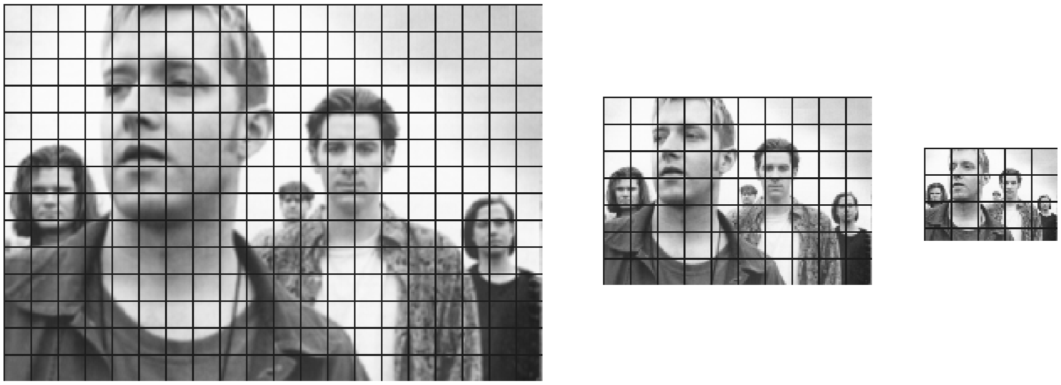


Figure 5: The two parts of the outer loop are depicted above. The original image, on the left, is downsampled once (middle image) and twice (right image). The scale of the smallest face is less than ten and hence this face is not detected. The next three in size are in the scale range $10 - 20$ and should be detected in the left image and the biggest face should be detected in the middle image.

From left to right, the third (middle) face is too small to be detected; the first, fourth and fifth faces are in the scale range $10 - 20$ and therefore we expect to detect them in the left image; the second face is in the range $20 - 40$ and we expect to detect it in the middle image.

The heart of the detection algorithm, the inner loops, is the search for a face in an image $I \in \mathcal{I}$ with pose in Θ. For each cell $\Lambda \in \mathcal{C}$, the learning routine (see below) yields an invariant detector f_Λ. The final detector, call it $F : \mathcal{I} \to \{0, 1\}$, depends only on the binary values $\{f_\Lambda, \Lambda \in \mathcal{C}\}$: $F(I) = 1$ if and only if there is a "chain of ones" - a complete sequence of positive responses among the $\{f_\Lambda, \Lambda \in \mathcal{C}\}$ ranging from the coarsest cell $\Lambda_{0,1} = \Theta$ down to one of the finest cells. In other words, there is a sequence $\{\Lambda_{m,l_m}, m = 0, ..., M\}$ with $\Lambda_{m+1,l_{m+1}} \subset \Lambda_{m,l_m}$ such that $f_\Lambda(I) = 1$ for each such $\Lambda = \Lambda_{m,l_m}$.

However, we do not evaluate $F(I)$ by first computing every $f_\Lambda(I)$ and then checking for a chain of ones. This would be highly inefficient. Instead, among all sequential procedures for evaluating F, we take the one which minimizes the average amount of

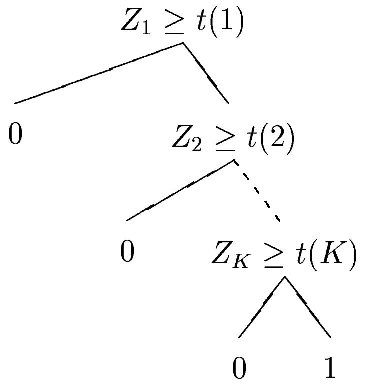


Figure 6: The function $Z_k(I)$ is the number of conjunctions of size k found in the image I. Instances of clutter and faces are separated by progressively checking for at least $t(k)$ conjunctions of size of k. Many subimages can be immediately dismissed as object candidates based on edge counts alone (Z_1); more global confusions require further examination involving increasingly structured edge arrangements.

computation under a certain model for the computational cost and the joint probability distribution (under P_0) of the random variables $\{f_\Lambda, \Lambda \in \mathcal{C}\}$.

Finally, each detector f_Λ embodies a coarse-to-fine progression in feature complexity. The features are conjunctions of disjunctions of edge fragments; the complexity is the size of the conjunction. "Tests" of every complexity $k = 1, ..., K$ must be verified in order to continue processing. Thus, each f_Λ has the form of a right vine (Figure 6) proceeding from $k = 1$ down to $k = K$, just as in checking for Napoleon. Verifying a test of complexity k means finding at least $t(k)$ conjunctions (decomposable arrangements) of size k: see §6.

5.2 Learning

Whereas f_Λ is defined explicitly (in §6) in terms of a P_Λ-dependent family of random variables on $\mathcal{I}$, the actual construction is inductive, based on a sample of training images of faces with a pose in Λ. Up to translation and reflection, there is one learning problem for each cell Λ in the decomposition of Θ. In other words, if one cell can be shifted or reflected to another then obviously we simply shift or reflect the tests. Thus, with our decomposition (three times quaternary in location and one time binary in scale and tilt), there are seven separate learning problems; these are the cells in Table 1 modulo reflection around the vertical axis.

The learning might be simplified by "scaling" the tests dedicated to one cell in order to construct tests for another cell with a different range of scales but otherwise equivalent. We have not done this. In the limit, one could train only at a reference pose and then attempt to transform the tests to accommodate any given subset Λ of poses. Despite the reduction in the amount of training, there are disadvantages. How does one transform the tests so as to maintain both efficiency and discrimination power? We have not explored the tradeoffs.

We induce features and estimate thresholds based on the empirical measure $\hat{P}_\Lambda$ generated by a training set $\mathcal{L}_\Lambda$. By and large, training amounts to estimating the probability distribution under P_Λ of image events, i.e., calculating relative frequencies in $\mathcal{L}_\Lambda$; these estimates determine the components of f_Λ. The training set $\mathcal{L}_\Lambda$ is assumed to be a random sample from $\mathcal{I}$ under P_Λ. An important constraint is that the size of $\mathcal{L}_\Lambda$ would not be sufficiently large to reliably estimate a number of *interdependent* parameters of the same order as the number we estimate.

6 Features

Throughout this section, we fix a pose cell $\Lambda \in \mathcal{C}$. A *test* is a binary function on $\mathcal{I}$. We will define a hierarchy of tests, from simple and localized to more complex and more spatially extended, whose statistics in the two populations $\mathcal{I}_F$ and $\mathcal{I}_B$ become increasingly disparate. In §6.1 we define "elementary tests" X_i, which represent localized edge fragments and involve comparisons of intensity differences; then, in

§6.2, we consider conjunctions

$$X_A = \prod_{i \in A} X_i$$

of elementary tests, which represent spatial arrangements of edge fragments.

Define $\delta_t(u) = 0$ if $u < t$ and $\delta_t(u) = 1$ if $u \geq t$. The detector f_Λ dedicated to Λ is then:

$$f_\Lambda \;=\; \prod_{k=1}^{K(\Lambda)} \delta_t \left(\sum_{A \in \mathcal{A}} X_A \right) \tag{1}$$

where $t = t(\Lambda, k)$ is a threshold and $\mathcal{A} = \mathcal{A}(\Lambda, k)$ represents a distinguished family of conjunctions of size k dedicated to poses in Λ. The particular conjunctions $A \in \mathcal{A}$ are the "decomposable" ones mentioned earlier. As we shall see, the difference in likelihood of the events $\{X_A = 1\}$ on faces and general backgrounds grows quickly with $k = |A|$. This property is pivotal in reducing the sums to manageable size (order 100), thereby "summarizing" a large disjunction of conjunctions.

6.1 Elementary Tests

An *elementary test* is a local disjunction of local filters. In our experiments the local filters detect edge fragments; other, more sophisticated, filters might be more effective. The edge filter we use is described in (Amit & Geman 1999) and additional details may be found in (Fleuret 2000). Briefly, the filter is applied at each location in G, and has an direction (horizontal, vertical, and two diagonals) and a contrast (positive or negative), yielding eight "types" denoted by $\xi = 1, ..., 8$. For example, in the case of a horizontal edge "at" (u, v), the absolute difference $|I(u, v) - I(u, v+1)|$ is compared with a threshold, with the differences $|I(u, v) - I(u', v')|$ for the nearest neighbors (u', v') of (u, v) and with the differences $|I(u, v+1) - I(u', v')|$ for the nearest neighbors (u', v') of $(u, v+1)$; it has positive contrast if $I(u, v) > I(u, v+1)$. The definitions of the other filters are analogous.

The principal motivation for using comparisons of intensity differences is to gain a measure of photometric invariance. One major difficulty in detecting faces is the

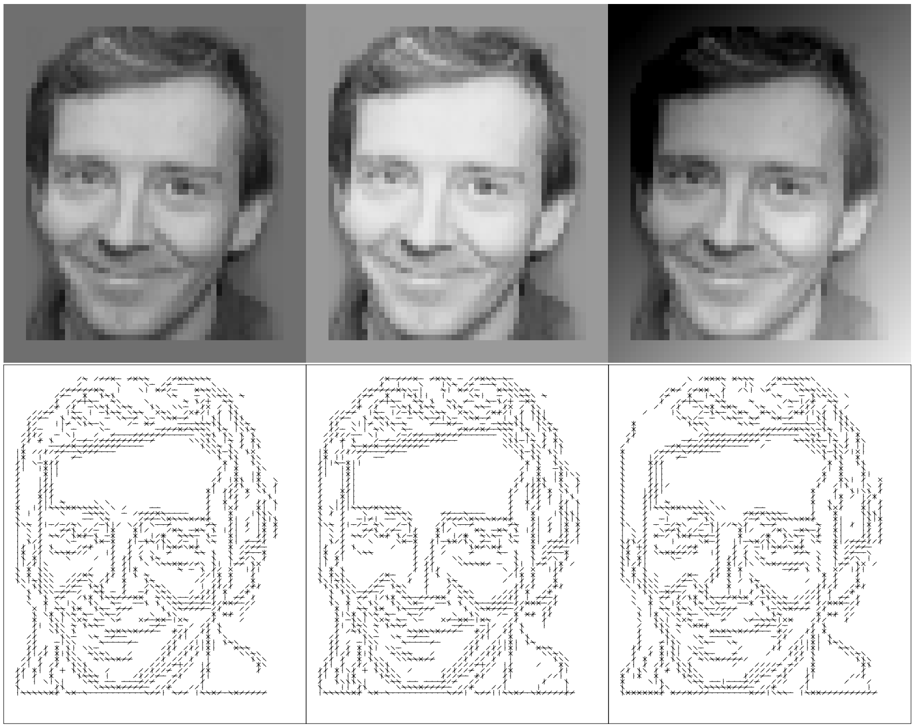


Figure 7: Detected edges on a training face under three illuminations.

variation in the appearance of faces due to the vagaries of lighting; see for example the discussion in (Ullman 1996). In order to diminish the variation, methods such as those based on neural networks usually require preprocessing (Rowley 1999), for instance subtracting a linear component from the grey level map followed by histogram equalization (Sung & Poggio 1998), which can be costly. Instead, the information we extract from the greylevels are comparisons of intensity differences, which are invariant to linear transformations of the greyscale. In Figure 7 we show three versions of a training face together with the detected edges.

There is an one elementary test $X = X(I)$ for each location (u, v), each filter type ξ and each "tolerance" $\eta = 1, 2, ..., 10$. Then $X = 1$ if there is an edge of type ξ at any location along a line of length η centered at (u, v) and orthogonal to the filter direction; otherwise $X = 0$. Thus, for example, in the case of a positive, horizontal type at location (u,v) and tolerance $\eta = 3$, the test $X = 1$ if there is an horizontal edge with positive contrast at at least one of the locations $\{(u,v-1),(u,v),(u,v+1)\}$; see (Fleuret 2000) for more details.

The tolerance parameter η is crucial for achieving a degree of invariance to small geometric deformations of the intensity surface. *It allows the elementary tests to be adapted to the generality of the pose.* The larger is Λ, the more the edges need to "float" in order to capture a reasonable percentage of object presentations. Specifically, for each cell Λ, we only consider elementary tests for which

$$P_\Lambda(X = 1) \geq 0.5. \tag{2}$$

These probabilities are estimated from $\mathcal{L}$; in other words we require $X(I) = 1$ for at least fifty percent of the training faces I with a pose in Λ. In addition, we then suppress other elementary tests of the same type and location with a tolerance larger than η, which necessarily also satisfy the constraint, thereby keeping only the minimal tolerance achieving a fifty percent incidence. Let $\{X_1, X_2, ..., X_N\}$ denote the surviving elementary tests, where $N = N(\Lambda)$.

6.2 Decomposable Tests

We refer to a subset $A \subset \{1, \ldots, N\}$ as an *arrangement* since it determines a set of approximate locations (and orientations) in the grid G corresponding to the elementary tests $X_i, i \in A$. Then $X_A = 1$ if and only if $X_i = 1$ for each $i \in A$, a spatial conjunction of elementary tests. Let $suppX_i \subset G$ be the set of η edge locations which appear in the definition of X_i. In order to limit the family of arrangements we shall assume that $suppX_i \bigcap suppX_j = \emptyset$ whenever $i, j \in A$ and $i \neq j$. We write $|A|$ for the size of A. The family $\{X_A\}$ is our pool of features; the classifier will be constructed from a subset of these - the decomposable ones - as indicated in (1).

We want to find arrangements A for which the statistics of X_A are as different as possible under P_0 and P_Λ. Since estimation under P_0 is problematic (see §10),

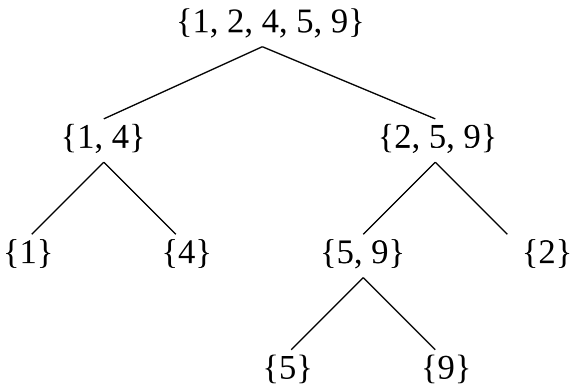


Figure 8: A test is ρ-decomposable if it can be broken down in at least one way into positively correlated subarrangements.

we will attempt to obtain the desired disparity by constructing arrangements which are large but still likely under P_Λ. Size alone renders them rare under P_0. The construction is based on correlation. Let $\rho(U, V)$ denote the correlation coefficient of random variables U and V with respect to P_Λ. For binary variables with $0 < P_\Lambda(U = 1), P_\Lambda(V = 1) < 1$ we have

$$\rho(U, V) = \frac{P_\Lambda(U = 1, V = 1) - P_\Lambda(U = 1)P_\Lambda(V = 1)}{(P_\Lambda(U = 1)P_\Lambda(U = 0)P_\Lambda(V = 1)P_\Lambda(V = 0))^{1/2}}.$$

Consider arrangements $X_i X_j$ of size two. We could filter all such pairs by requiring that $\rho(X_i, X_j) \geq \rho$ for some threshold ρ, $0 < \rho < 1$. This yields pairs of elementary tests which tend to occur (or not occur) together on objects. Similarly, $X_i X_j X_k$ might be a good candidate for a discriminating arrangement of size three if, in addition, $\rho(X_i X_j, X_k) \geq \rho$. Continuing in this way, we can single out arrangements of size four by combining two "good" pairs $X_i X_j$ and $X_k X_l$ and further requiring that $\rho(X_i X_j, X_k X_l) \geq \rho$. And so forth.

Define a *decomposition* of A to be any nested set of binary partitions (i.e., successive binary refinements) all the way down to individual elements of $\{1, 2, ..., N\}$. We shall also assume that a partition element splits evenly if its size is even and splits into two child elements whose sizes differ by exactly one if its size is odd.

Call it a *ρ-decomposition* if the correlation inequality holds at every split. In Figure 8 we show one decomposition of $A = \{1, 2, 4, 5, 9\}$. It is a ρ-decomposition if $\rho(X_1X_4,\ X_2X_5X_9) \geq \rho$, $\rho(X_1,\ X_4) \geq \rho$, $\rho(X_5X_9,\ X_2) \geq \rho$ and $\rho(X_5,\ X_9) \geq \rho$. Finally, an arrangement A, or the corresponding test X_A, will be called *ρ-decomposable* if there is *at least one* ρ- decomposition of A. Summarizing,

Definition: *A test X_A is ρ-decomposable if it is an elementary test or if exists two ρ-decomposable tests X_B and X_C with*

- $A = B \cup C, \quad B \cap C = \emptyset$
- $||B| - |C|| \leq 1$
- $\rho(X_B, X_C) \geq \rho$

6.3 A Likelihood Bound

In general $P_0(X_A = 1)$ and $P_\Lambda(X_A = 1)$ depend on A and decrease as $|A|$ increases. A reasonable assumption for P_0 is some type of exponential decrease, and indeed this is what we observe empirically. On the other hand, if X_A is ρ-decomposable, we should expect a slower rate of decrease under P_Λ. This is certainly what we observe experimentally; see Figure 9. In fact, the rate of decrease is $\rho^{\log_2 k}$. As a result, for "reasonable" values of ρ, $P_\Lambda(X_A = 1) \gg P_0(X_A = 1)$ for "large" A. We cannot say anything precise about the likelihood ratio since we do not propose a model for P_0. But we can give lower bounds on $P_\Lambda(X_A = 1)$. Let $\mathcal{A}(\Lambda, k, \rho)$ denote the set of all ρ-decomposable arrangements with $|A| = k$.

Two bounds are easy to obtain. One is

$$P_\Lambda(X_A = 1) \geq \left(\min_{1 \leq i \leq N} P_\Lambda(X_i = 1) \right)^k \tag{3}$$

which results directly by iterating the basic inequality that defines decomposability. Another is $P_\Lambda(X_A = 1) \geq U(k)$, obtained numerically and recursively from

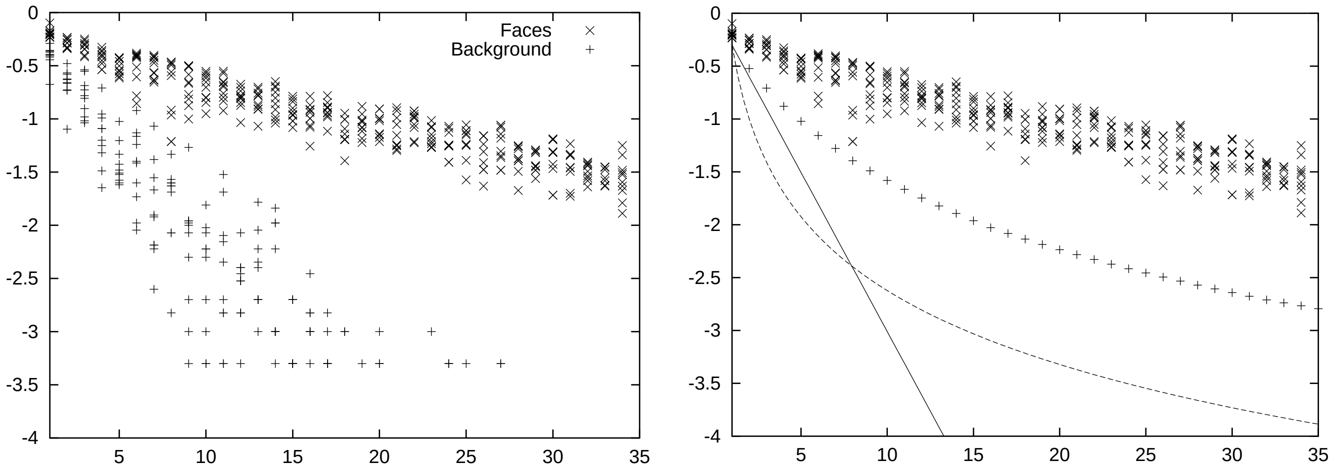


Figure 9: The empirical behavior of randomly selected decomposable tests. The vertical axis is log-probability and the horizontal axis is complexity (k). Left: Estimated probabilities on face and background subimages. Right: Three lower bounds: numerical U (+++), analytical (4) (dashed line), exponential (3) (solid line).

- $U(0) = \min_{1 \leq i \leq N} P_\Lambda(X_i = 1)$
- $U(2k) = \rho \cdot U(k) \cdot (1 - U(k)) + U(k)^2$
- $U(2k+1) = \rho \cdot \sqrt{U(k) \cdot (1 - U(k)) \cdot U(k+1) \cdot (1 - U(k+1))} + U(k) \cdot U(k+1)$

There is no analytic expression for U.

A closed-form bound which is larger (and hence better) than the exponential bound is given below. We will assume that $P_\Lambda(X_A = 1) \leq 0.5$ for every $A \in \mathcal{A}(\Lambda, k, \rho)$. This is implied by $P_\Lambda(X_i = 1) \leq 0.5$, which is the case in practice if we replace the value 0.5 in (2) by one slightly smaller because, due to the tolerance parameter, the probabilities in (2) cluster tightly just above the threshold.

Theorem 1: *For any $k \geq 1$, $\rho > 0$ and $A \in \mathcal{A}(\Lambda, k, \rho)$,*

$$P_\Lambda(X_A = 1) \geq \min_{1 \leq i \leq N} P_\Lambda(X_i = 1) \cdot \rho^{\log_2 k}. \tag{4}$$

In Figure 9 we display the shape of these bounds as well as the empirical behavior of tests. For each k, there are ten estimated values of $P_\Lambda(X_A = 1)$ for ten tests X_A randomly sampled from thousands learned from training data; see §7. The estimates are relative frequencies in training data. As can be seen, the bound in (4) captures the actual rate of decrease fairly well.

6.4 Progression in Feature Complexity

As indicated earlier, we implement f_Λ as the series of filters defined in (1) and depicted in Figure 6. Each filter is applied only when all simpler ones have rejected background. Since the overwhelming majority of subimages examined are in fact background, very few are investigated in detail. As seen in (1), the filter of complexity k is

$$Z_{\Lambda,k}(I) = \sum_{A \in \mathcal{A}(\Lambda,k,\rho)} X_A(I),$$

the number of ρ-decomposable tests of size k which are positive on I.

For simplicity, we fix ρ and suppress it from the notation. In theory, the optimal value is the one which minimizes the false positive rate of f_Λ but we have not performed any systematic exploration of the possible values, or even considered allowing ρ to depend on Λ. In all experiments we take $\rho = 0.1$ for every pose cell.

The maximum size K and the thresholds $t(1), ..., t(K)$ are determined as follows. Let K be the largest k which "covers" the object class in the sense that $P_\Lambda(Z_{\Lambda,k} \geq 1) = 1$. (In our experience it never happens that arrangements of size k cover but arrangements of size $j < k$ do not.) Given thresholds $t(1), \ldots, t(K)$, and according to (1), we classify I as object if it contains more than $t(k)$ ρ- decomposable tests of size k for each $k = 1, ..., K$. The thresholds $t(1), ..., t(K)$ are defined by

$$t(k) = \max\{j : P_\Lambda(Z_{\Lambda,k} \geq j) = 1\}. \tag{5}$$

In other words, the thresholds are the maximum values which preserve the hard constraint that $\alpha(f_\Lambda) = 0$.

There are several practical obstacles to implementing the detectors f_Λ exactly as defined in the previous section.

- We don't have $\mathcal{A}(\Lambda, k, \rho)$. This would require far more precise information about P_Λ than can be gleaned from any training set. Also, the family is too large to enumerate. Instead we will estimate a fixed number of decomposable tests of each size, basing correlation estimates on $\mathcal{L}$.

- The thresholds are difficult to estimate directly from $\mathcal{L}$ without overfitting. In the following section we shall indicate how this can be accomplished by synthetically enlarging the training set. This also solves the problem of having enough data to estimate correlations for fine pose cells.

- If a subset of decomposable tests is selected based on likelihood alone, the test locations will concentrate on certain regions of the object and be highly redundant, as well as provide no protection against occlusion. Consequently, for each k, we force the decomposable tests to "spread out" by restricting the number of times each original edge appears in an arrangement.

7 Feature Learning

Assume Λ is still fixed and let $\mathcal{L}_\Lambda$ be the set of training images with pose in Λ. Most of the images in $\mathcal{L}_\Lambda$ are obtained synthetically by transforming images in the original training set $\mathcal{L}$. Bearing this in mind, in order to simplify the notation we shall simply write $\mathcal{L}$ for $\mathcal{L}_\Lambda$ and $\mathcal{A}(k)$ for $\mathcal{A}(\Lambda, k, \rho)$, the set of all ρ-decomposable arrangements of size k, as defined in §6.3. One goal of learning is to estimate a subfamily of $\mathcal{A}_\mathcal{L}(k) \subset \mathcal{A}(k)$ of size n for each $k \leq K$. The other learning task is to estimate the thresholds $t(1), ..., t(K)$.

Whereas the definition of a decomposable products is top-down, the production of examples is bottom-up. Correlations are estimated under $\hat{P}_\Lambda$, the empirical measure derived from $\mathcal{L}$ ($\mathcal{L}_\Lambda$). The construction is recursive: First build a family $\{X_iX_j\}$, then a family $\{X_iX_jX_k\}$, etc. In order to construct decomposable products of size $2k$ we only need those of size k, and to construct those of size $2k+1$ we only need those of sizes k and $k+1$.

Eventually, we want tests $\{X_A, A \in \mathcal{A}_{\mathcal{L}}(k)\}, k = 1, ..., K$, with various properties.

- First, they should "cover the population" in the sense that, for every face image, at least one test of each complexity is positive. In other words, $t(k) \geq 1$ for each $k = 1, ..., K$, where $t(k)$ is defined in (5). (Of course the probability in (5) is estimated from $\hat{P}_\Lambda$.)

- Second, they should be "spatially non-redundant," in the sense of having supports spread out over the image plane. This does not occur naturally; indeed, without some constraint, the locations of the tests tend to accumulate on certain areas of the face.

- Third, there should be relatively few tests. Specifically, the sums appearing in (1) should be of order 100; otherwise, we lose computational efficiency. Indeed, having a "small" number of decomposable tests with the two properties above implies a large degree of invariance.

For each k we first generate a very large family $\mathcal{F}(k)$ of decomposable tests and then select a subset $\mathcal{F}^\circ(k) \subset \mathcal{F}(k)$ of size N by random sampling subject to the first two constraints mentioned above. The final set, $\mathcal{A}_{\mathcal{L}}(k)$, is a small subset of $\mathcal{F}^\circ(k)$. This multi-step procedure is how we generate a family which is sufficiently rich to contain a smaller subfamily which has all the desired properties.

Consider the even case. The large family $\mathcal{F}(k)$ is the set of all arrangements $A_1 \bigcup A_2$ where

- $A_1, A_2 \in \mathcal{F}^\circ(k)$;

- $\hat{\rho}(X_{A_1}, X_{A_2}) \geq \rho$;

- $suppX_{A_1} \bigcap suppX_{A_2} = \emptyset$.

Here, $suppX_A = \cup_{i \in A} suppX_i$. The process is initialized with $\mathcal{F}^\circ(1)$, the family of distinguished elementary tests described in §6.1. If the covering condition for the elementary tests fails, then we do not attempt to build a classifier at the level of generality of Λ. For instance, the covering condition fails if the location of the face is allowed to roam over a 32×32 block (and scale and tilt are unrestricted). This is why we begin at the 16×16 level. The process terminates when it is impossible to satisfy the constraints. Generally, $N \ll |\mathcal{F}(k)| \ll N^2$. The exact sampling procedure for choosing $\mathcal{F}^\circ(k) \subset \mathcal{F}(k)$ and then $\mathcal{A}_{\mathcal{L}}(k) \subset \mathcal{F}^\circ(k)$ is described in (Fleuret 2000).

Ghe natural estimators of the thresholds $t(1), ..., t(K)$ are

$$\hat{t}(k) = max \left\{ t : \hat{P}_\Lambda \left(\sum_{A \in \mathcal{A}_{\mathcal{L}}(k)} X_A \geq t \right) = 1 \right\}, \quad k = 1, ..., K.$$

Due to the synthetic deformations of the original training faces, these thresholds are actually very conservative and can be used in practice as defined.

Finally, by construction, the tests in $\mathcal{A}_{\mathcal{L}}$ are ρ-decomposable with respect to $\hat{P}_\Lambda$. Are they ρ-decomposable with respect to P_Λ? It appears that some are not and some are at even a larger value of ρ. Let $\rho_0 = .1$; this is the value used in our experiments. Recall that each constructed $A \in \mathcal{A}(\Lambda, k)$ has a *proposed* ρ_0-decomposition. One can then use additional data to verify this decomposition by re-estimating the correlations. Further, one can determine $\rho_{max}(A)$, the maximal value of ρ for which the given decomposition of A is a ρ-decomposition. This value may be smaller or larger than ρ_0. Some results are reported in (Fleuret 2000). For example, in one typical experiment, the proposed decompositions for about 95% of the arrangements are valid at $\rho > 0$, 80% at $\rho \geq .1$ (the target value) and 45% at $\rho \geq .2$. These estimates are conservative because the arrangements could decompose differently.

8 Sequential Testing

Recall that the exploration of poses is based on a sequence of nested partitions of Θ corresponding to divisions on location, scale and tilt. We declare a face with pose in Θ if and only if we confirm at least one decreasing sequence of pose cells arriving at a fine cell. We use a tree-structured strategy for checking this condition. Roughly speaking, the tests $\{f_\Lambda, \Lambda \in \mathcal{C}\}$ are performed adaptively in the order which would minimize the mean amount of computation (under the background hypothesis) necessary to determine F under a certain statistical model described in Appendix C. That particular adaptive procedure, "the coarse-to-fine tree," is the topic of this section.

Let $\gamma(j)$ denote the set of ancestors of the fine cell $\Lambda_{M,j}, j = 1, ..., L_M$:

$$\gamma(j) = \{(m,l) : \Lambda_{M,j} \subset \Lambda_{m,l}\}.$$

The detector f_Λ corresponding to cell $\Lambda = \Lambda_{m,l}$ will be denoted by $f_{m,l}$. *Then $F(I) = 1$ if and only if $I \in \Gamma$, where*

$$\Gamma = \{I \in \mathcal{I} : \exists j \ni \ \forall (m,l) \in \gamma(j) \ f_{m,l}(I) = 1\}. \tag{6}$$

This characterizes F but does not describe an algorithm for evaluating it. The particular algorithm for checking the condition $I \in \Gamma$ is what we refer to as the testing strategy and is described below.

Under very mild assumptions (see Appendix B), *any* detector f based entirely on the filters $\{f_\Lambda, \Lambda \in \mathcal{C}\}$ has *overall* false negative error zero (i.e., with respect to $P_1 = P_\Theta$) if and only if $f(I) = 1$ for every $I \in \Gamma$. Consequently, among all such detectors, the smallest false positive error is achieved by $f = F$.

We describe the testing strategy for a binary decomposition of Θ ($L_m = 2^m$). The general case is the same but the diagrams are messy. Let $\mathcal{T}$ be the family of all labeled trees which evaluate F. Each $T \in \mathcal{T}$ is a variable-depth binary tree with each internal node labeled by a test in $\{f_{m,l}\}$ (the same test may appear more than once) and each

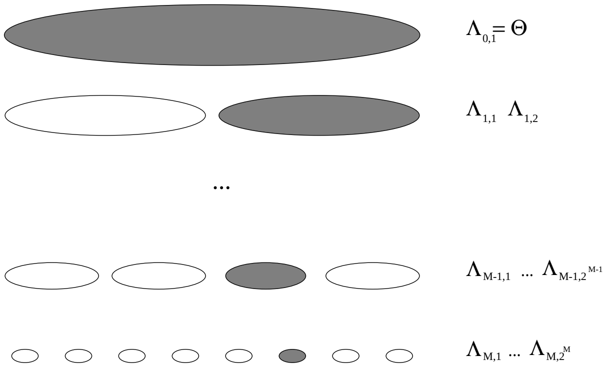

$\Lambda_{0,1} = \Theta$
$\Lambda_{1,1}$ $\Lambda_{1,2}$
...
$\Lambda_{M-1,1}$ ... $\Lambda_{M-1,2^{M-1}}$
$\Lambda_{M,1}$ ... $\Lambda_{M,2^M}$

Figure 10: A binary decomposition of pose space and a "chain of ones" indicated in grey.

external node (leaf) is labeled either "0" or "1". The left (respectively, right) branch emanating from an internal node labeled by $f_{m,l}$ indicates $f_{m,l} = 0$ (resp., $f_{m,l} = 1$).

Overloading the symbol T, we will also write $T(I)$ for the corresponding detector: $T(I) = 0$ (resp. $T(I) = 1$) if sending I down the tree leads to a "0" (resp. "1") leaf. *In order to represent F, $T(I) = 1$ if and only if $I \in \Gamma$.* This means that a leaf t is labeled "1" if and only if, for some $j = 1, ..., L_M$, the history of tests along the branch from t to the root contains the event $\{f_{m,l} = 1 \ \forall (m,l) \in \gamma(j)\}$. See Figure 10. Equivalently, a leaf t is labeled "0" if and only if there is a covering partition of "0" tests, i.e., the leaf history contains an event of the form $\{f_{m_r,l_r} = 0, r = 1, ..., R\}$ where $\cup_r \Lambda_{m_r,l_r} = \Theta$.

Of the many trees in $\mathcal{T}$, the least efficient simply performs all the tests in some fixed order along every branch and therefore has depth uniformly equal to $\sum_{m=0}^{M} L_m$. Another procedure is the "depth-first, coarse-to-fine" tree T^*. It is depicted in Figure 11 and Figure 12 for the two cases $M = 1$ and $M = 2$, and can be defined recursively, as indicated in Figure 13. It is unique up to a permutation of the testing order within each layer, which has no significance. *The tree T^* is the representation of the detector used by the algorithm.* It is efficient because no finer test (along a chain) is ever performed before all coarser ones have failed to eliminate a candidate subimage, and

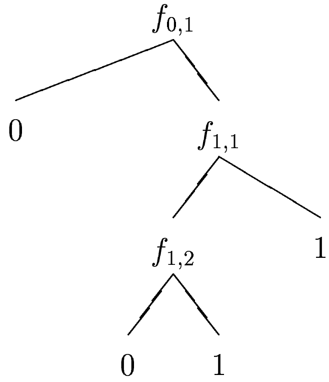


Figure 11: The coarse-to-fine tree T^* for $M = 1$.

the testing is stopped when F is determined. Notice that the visitation of cells is not strictly coarse-to-fine along every branch of the tree, i.e., there is "backtracking" up the pose hierarchy.

In Appendix C we present a model for the statistical distribution of the tests $\{f_\Lambda, \Lambda \in \mathcal{C}\}$ with respect to P_0, as well as their cost structure. Let $\mathcal{H}$ denote this set of hypotheses and let $E_0C(T)$ denote the expected cost of $T \in \mathcal{T}$ under P_0 (see Appendix C). Then

Theorem 2: *Under $\mathcal{H}$, the coarse-to-fine tree minimizes computation:*

$$E_0C(T^*) = \min_{T \in \mathcal{T}} E_0C(T).$$

Notes: i) In an earlier version of this paper, this result was stated as a "conjecture." It has since been proven in collaboration with Franck Jung. The proof, which is rather complex, will appear elsewhere.
ii) In processing real scenes, the algorithm based on T^* is in fact considerably faster than various alternatives, such going straight to the fine cells, in which case the processing image corresponding to Figure 1 is much flatter (Fleuret 2000).

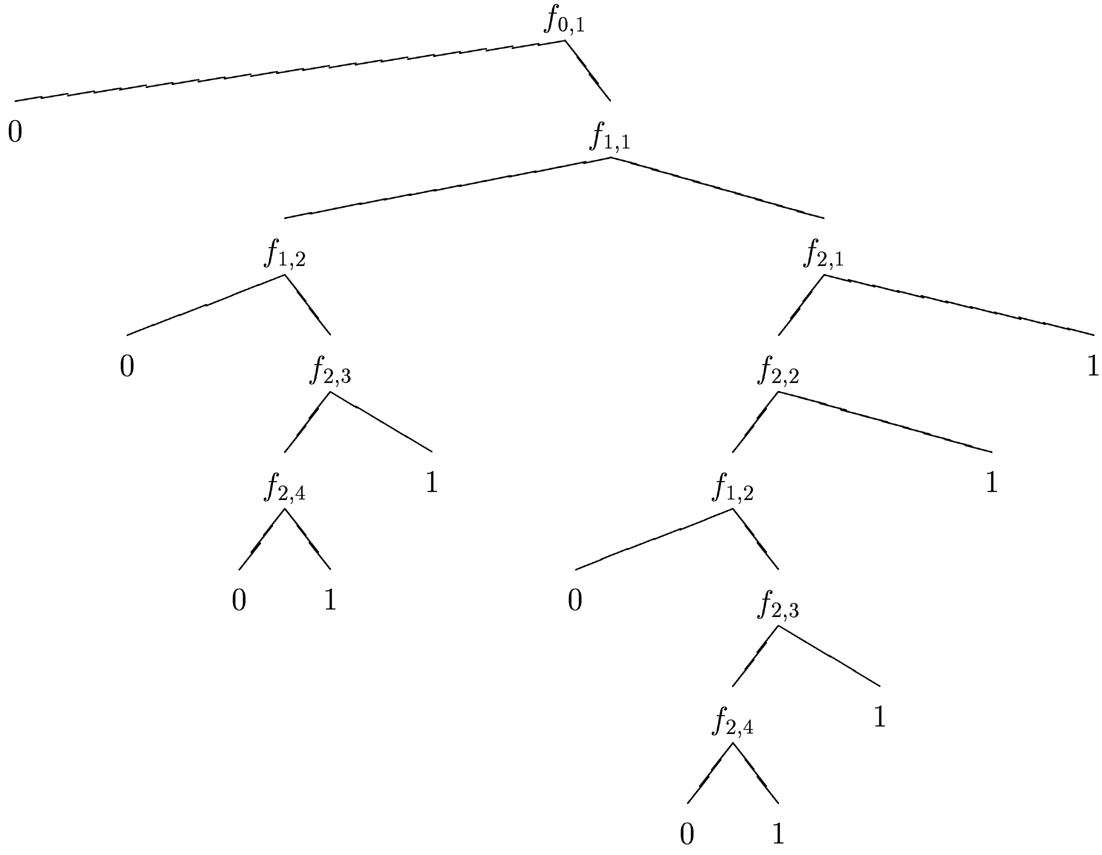


Figure 12: The coarse-to-fine tree T^* for $M = 2$.

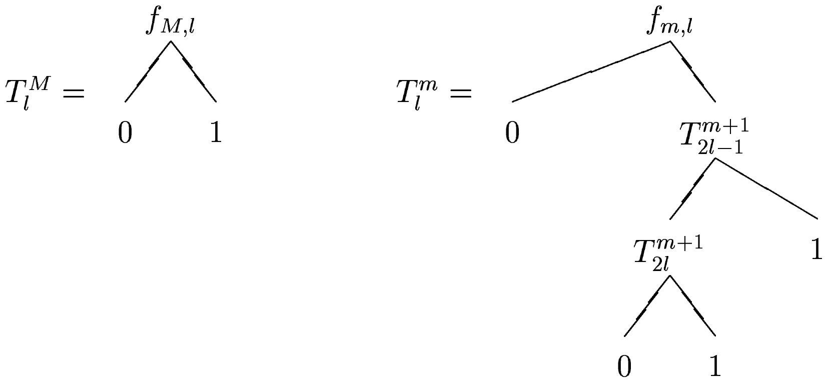


Figure 13: Recursive definition of T^*.

9 Experiments in Face Detection

We have extracted 300 images from the Olivetti database of faces, corresponding to ten different frontal views of each of 30 individuals; this is $\mathcal{L}$. On each image, we have marked the locations of the eyes. This determines our three pose parameters - position, scale and tilt. The decomposition of Θ into pose cells was described in §3. To generate $\mathcal{L}_\Lambda$, i.e., training faces with a pose confined to Λ, we cannot simply use an appropriate subset of $\mathcal{L}$ since there will not be enough data for "small" cells. This is due to a limited sample of scales and tilts (we can always translate to any desired location). To overcome this, we synthesize a set $\mathcal{L}_\Lambda$ of size 1200: For each $I \in \mathcal{L}$ we select four poses from Λ at random (uniformly in position, scale, tilt) and then scale and rotate I to acquire each of these poses.

9.1 Learned Arrangements

Randomly chosen examples of learned arrangements of size eight are shown in Figure 14. The grey regions indicate the amount of disjunction in elementary tests. These arrangements are typical of the thousands inferred from $\mathcal{L}$. Generally, they utilize elementary tests based on edges in the region of the eyes, the mouth and the contours of the face.

One measure of the discriminating power of the tests was illustrated in Figure 9. Whereas we can build arrangements up to size 35, the maximum size $K(\Lambda)$ in the final detector is closer to 10 due to the covering criterion. We randomly sampled ten tests for each $k = 1, ..., 35$ and estimated the probability of a positive response given face (based on $\mathcal{L}$) and given background (based on randomly selected locations in natural scenes).

Figure 15 shows the estimated distributions of $Z_{\Lambda,k}$ under P_0 and P_Λ for $k = 5$ and $k = 8$. The possible values of $Z_{\Lambda,k}$ are $\{0, 1, ..., 100\}$ since $|\mathcal{A}_{\Lambda,k}| \equiv 100$. Finally, Figure 16 depicts an estimate of the function $k \to P_0(Z_1 \geq t(1), ..., Z_k \geq t(k))$, the

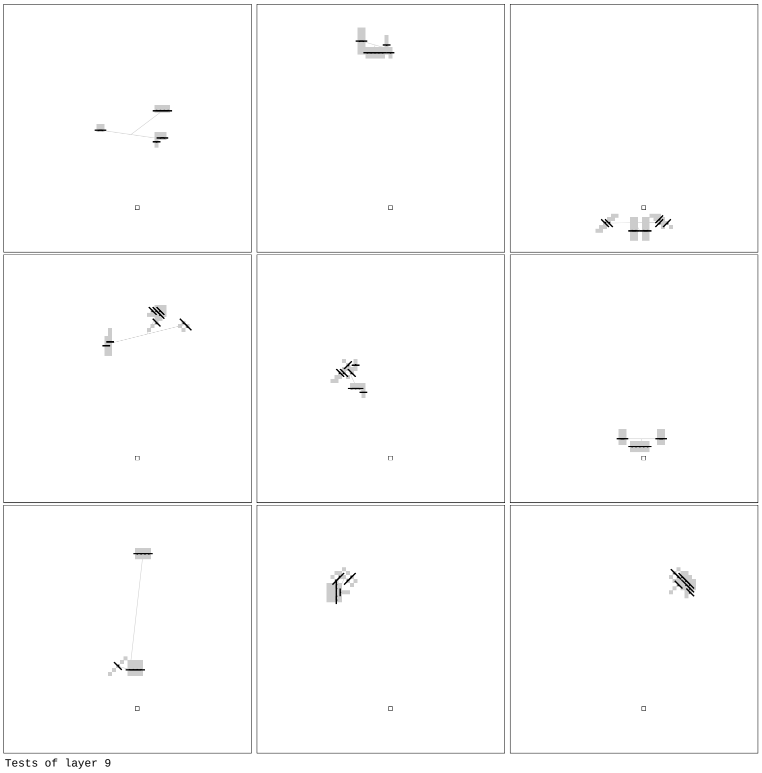


Figure 14: A random sample of learned decomposable arrangements of size nine. The shading indicates the amount of flexibility in the edge location.

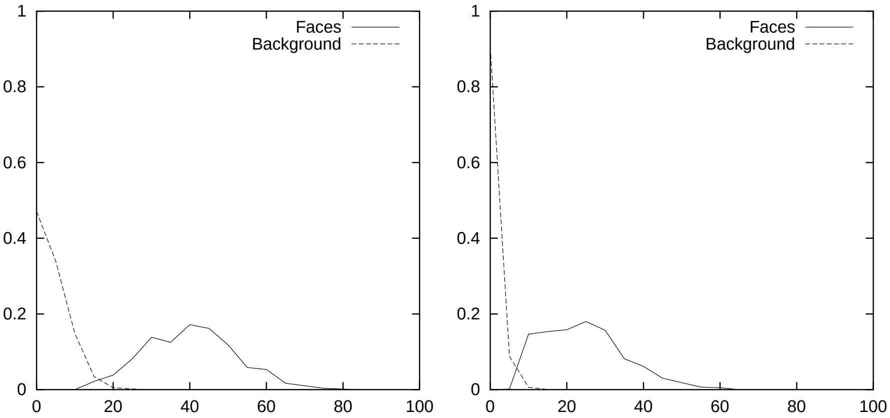


Figure 15: Estimated distributions of Z_5 (left) and Z_8 (right) on faces and background samples.

rate at which false positive error decreases with test complexity, shown as a solid line. The "+"s refer to the individual statistics $P_0(Z_k \geq t(k))$. The estimates are based on a large number of non-face images found on the WWW.

9.2 Processing Scenes

The search for a face at a reference pose terminates as soon as a chain of ones is found. Consequently, there is exactly one fine cell associated with each detection. However, given a face is present, the fine cell which is identified may be due to clutter in the vicinity of the face, and hence the precision of the detection is only reliable at the level of the coarsest cell. Still, the information in the fine cell is nearly always a very good guess at the pose. In our experiments, the coarsest cell restricts location to a 16×16 block; there is no restriction on tilt and no restriction on scale within the reference range, which means detecting scale in one of the ranges $10-20$, $20-40$, etc. The number of false positives is then the number of these coarse cells which are detected at some resolution and which do not contain a face.

We have tested the algorithm on several scenes collected from the WWW and

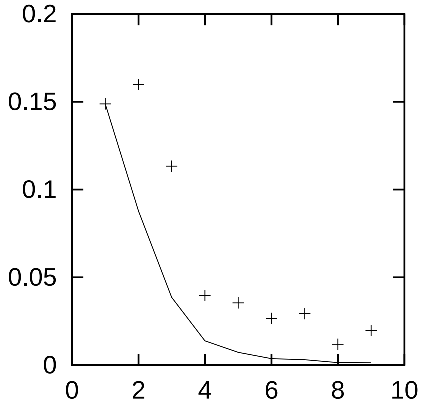


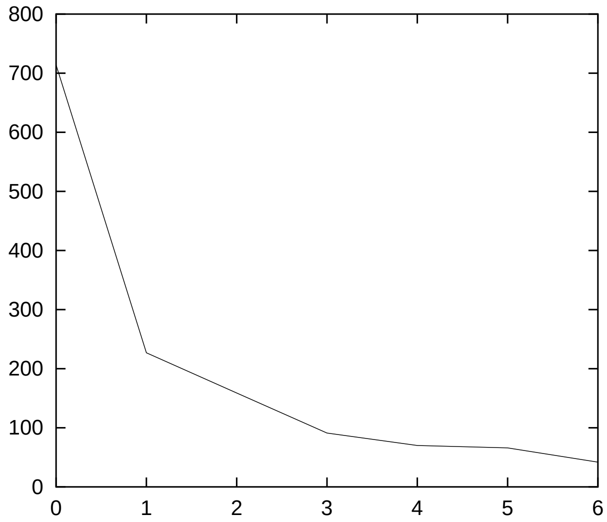
Figure 16: The rate of decrease in false alarms with text complexity.

from the set “C” of images collected at Carnegie Mellon University by H.A.Rowley et al (Rowley et al. 1998). One result appears in Figure 3. The scene is 450 × 380. The three faces which are about half-visible are missed. In Figure 17 we indicate the rate at which the number of alarms decreases during the focusing in pose, i.e., with the number of splits on the coarse cell. The value 714 in the righthand panel is the total number of 16 × 16 blocks in the image at all resolutions. Other results are shown in Figure 18 and Figure 19.

Measuring the amount of computation is not entirely straightforward. It depends on the scene, the computer, the source code and perhaps other factors. With a PC Pentium II (450Mhz), it takes about one-half second to process the scene in Figure 2; this is an average over 100 runs. Most of this time is spent on extracting the elementary tests; computing the detector F (at all resolutions) requires only about one-tenth of a second. Clearly, more efficient preprocessing would help.

9.3 Improvements

One fundamental limitation is that false detections often occur in areas of very high edge activity, as in foliage or fine textures. Indeed, nothing changes if edges are added



Depth	Number of alarms
	714
0	227
1	159
2	91
3	70
4	66
5	42

Figure 17: The number of alarms (detections) as a function of the depth m of focusing in pose space. The value corresponding to m is the number of blocks surviving past the the m'th partition.

to the vicinity of a region already labeled as a face. In order to remedy this flaw, we have done some preliminary experiments with "negative tests." We use exactly the same learning protocol and detection algorithm, except that we add elementary tests whose response is positive when the local filter response is negative everywhere in a strip orthogonal to the edge direction. We have also experimented with a finer pose decomposition, for instance splitting more than once on scale or tilt, and with more general notions of pose (see §3). Preliminary results are promising and suggest that many of the false positives can be eliminated.

9.4 Comparisons

It can be hazardous to compare the performance of one method with that of another. Still, due to the comprehensive analysis in (Rowley 1999) of publicly available images and to our familiarity with (Amit & Geman 1999), a few general statements appear evident. First, our false negative rate is smaller; a 15% rate is reported in (Rowley

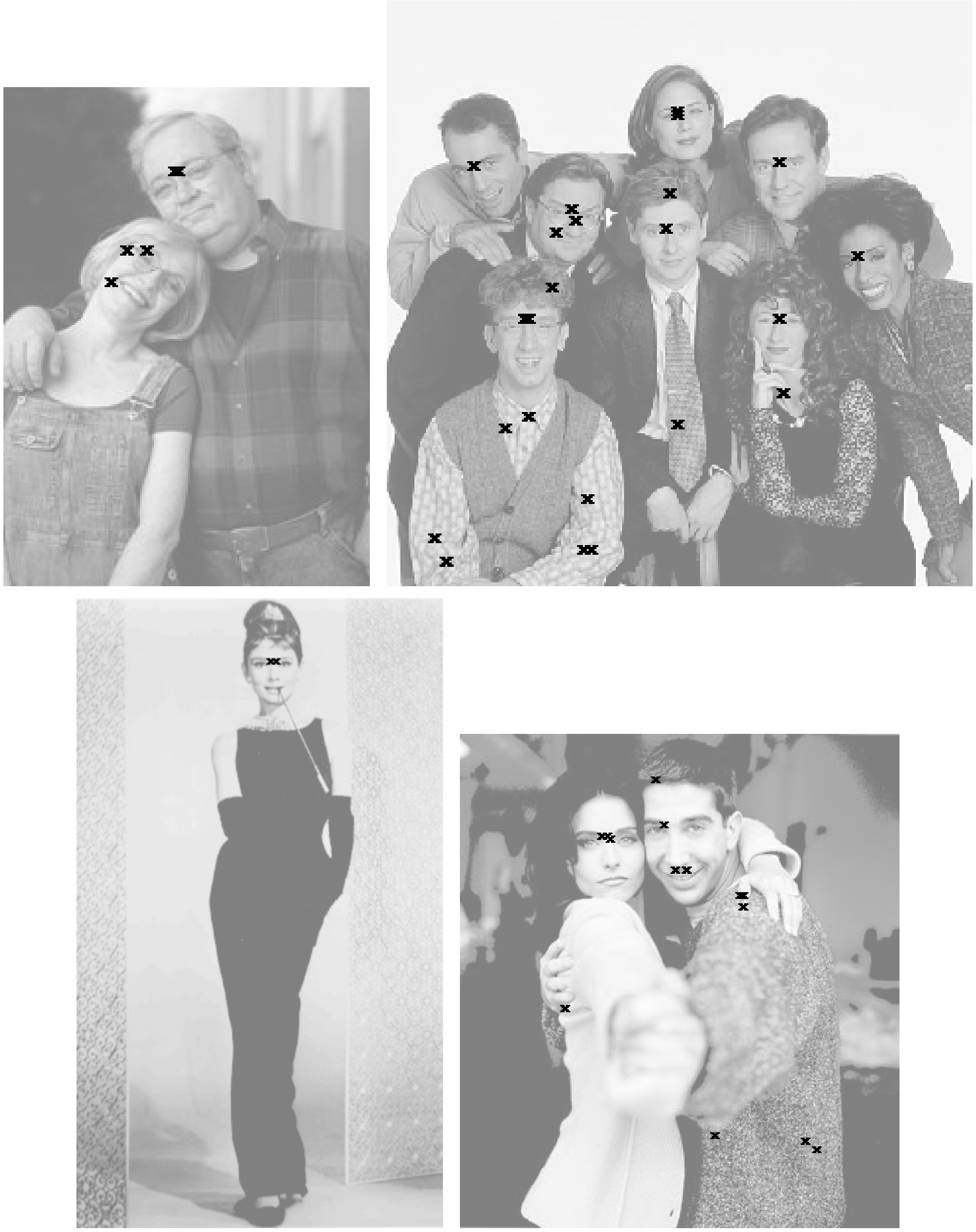


Figure 18: Additional results

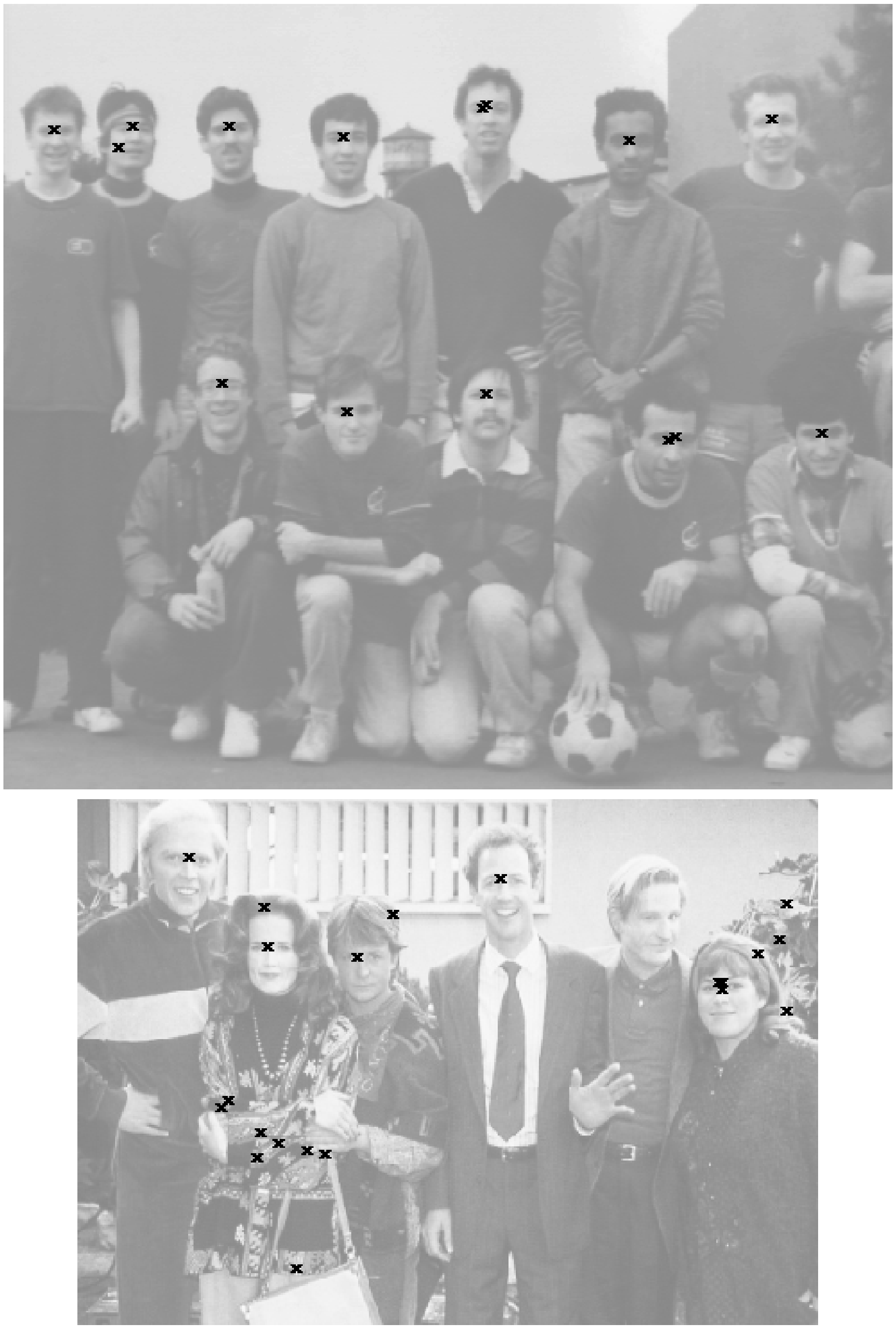


Figure 19: Additional results

1999) for an ensemble of images, and other authors (e.g., (Miao et al. 1999)) obtain similar rates. This is consistent with our formulation of the visual selection problem. Second, there seem to be fewer false alarms in (Rowley 1999). This statement is based on processing some of the same scenes as those analyzed in these references. It should be noted that no reported algorithm detects nearly all faces and nothing else. Our algorithm is faster than the one in (Amit & Geman 1999) and much faster than the one in (Rowley 1999), which requires $140s$ to process the scene in Figure 2 (with the PC mentioned earlier) and about $2s$ with a two-step, coarse-to-fine process for which the ensemble false negative rate climbs to 26%.

There are other measures of efficiency. The algorithm in (Amit & Geman 1999) is perhaps the simplest: The object representation is very compact and training only occurs at a reference pose, requiring only a few minutes as opposed to about an hour here and much longer in (Rowley 1999). Our face training set is the same as in (Amit & Geman 1999) and smaller than in (Rowley et al. 1998),(Sung & Poggio 1998). Finally, we often localize with less precision than some other algorithms. We could do better with more computation, for example by not terminating the search upon the first positive chain of responses; obviously there are many tradeoffs of this nature.

10 Discussion

We have argued that a good start on solving vision problems might be to think about computation, and this leads naturally to coarse-to-fine processing in several senses, including feature complexity and the search over nuisance parameters. Start with the simplest and most common properties over presentations, almost regardless of discriminating power; rejecting even a small percentage of background instances with cheap and universal tests is efficient. Then proceed to more complex and/or more dedicated properties, reserving any computationally intensive search for the very special confusions - those inevitable and diabolical arrangements of clutter which "look" like objects in the eyes of the features. Also, design the search to account for the fact that detecting an object at any given pose, or even localized set of poses, is an extremely rare event. We have illustrated these ideas with experiments on detecting frontal views of faces over a limited range of tilts and a large range of scales. Although there are certainly false alarms, the algorithm is fast and unlikely to miss a face.

This type of reasoning does not seem to drive the construction of very many vision algorithms, at least not in academic research. Instead, computation is usually an afterthought; for example, one seeks ways to speed up an algorithm originally motivated by other principles (deforming templates, the world is 3D, vision is compositional, inference should be Bayesian, etc.). Some notable exceptions include work on hashing (Lamdan, Schwartz & Wolfson 1988), Hough transforms (Rojer & Schwartz 1992),(Amit & Geman 1999), (Amit 1999), and tree-structured search (Grimson 1990), all of which have influenced our thinking.

Our treatment of features is statistical and inductive. We build a degree of invariance into elementary, binary features and then learn those conjunctions which are likely on object instances rather than having any other a priori distinguished property. The idea is to make the conjunctions "decomposable" relative to the statistics of the object class. The induction process does not utilize a background model (such as the minimax entropy model proposed in (Zhu, Wu & Mumford 1997)) or samples of backgrounds and confusions (as in (Sung & Poggio 1998) and (Rowley et al. 1998)), both of which might improve discrimination.

We have not appealed to general theories for hypothesis testing (for instance likelihood ratio tests based on models for P_0 and P_1) or for inductive learning (for instance structural risk minimization (Vapnik 1996)) or for feedforward classifiers ((Baum & Haussler 1989),(Devroye, Gyorfi & Lugosi 1995)). Instead, the global form of the detector is dedicated to the visual selection problem; also, each estimated parameter has an explicit interpretation (correlation or quantile) and is decoupled from the others, which renders training feasible without a large database. The generic component of the learning is the concept of a decomposable arrangement, which might be of interest in other domains; see (Fleuret 2000) for some remarks about natural language and cortical function.

How would this approach extend to detecting a truly three-dimensional object, or a more complex one (e.g., a cat) or to detecting many objects simultaneously? We don't know. Obviously there are more degrees of freedom in imaging a 3D or highly deformable object. But divide-and-conquer is a very powerful strategy, and can certainly be pushed a good deal further. Even in searching for a cat, perhaps enough efficiency can overcome the combinatorics - the sheer number of presentations and cat-like things - and more general pose hierarchies could be generated automatically based on feature counts. Compared with faces, many more confusions might be kept around for many more steps, and eliminating all of them might require on-line optimization and contextual analysis. However, since this would only occur in few places, detection would remain computationally efficient. As for detecting multiple objects, perhaps the key issue, at least in our framework, is "reusable parts" - representing different objects with the same arrangements whenever possible. For example, one might build a detector for a "new" object at some subset of poses from the detectors already built for other objects in various subsets.

Finally, in defense of limited goals, nobody has yet demonstrated that objects from even one generic class under constrained poses can be rapidly detected without errors in complex, natural scenes; visual selection by humans occurs within two hundred milleseconds and is virtually perfect.

Acknowledgements: We are grateful to Yali Amit for many suggestions during a running discussion of learning and invariance. The second author would also like to acknowledge the influence of unpublished work on coarse-to-fine machine vision with E. Bienenstock, S. Geman and D.E. McClure.

Appendix A: Proof of Theorem 1

Recall that the bound in question is $P_\Lambda(X_A = 1) \geq \min_{1 \leq i \leq N} P_\Lambda(X_i = 1) \cdot \rho^{\log_2 k}$. The result is evident for $k = 1$. Let $\xi = \min_{1 \leq i \leq N} P_\Lambda(X_i = 1)$ and let $\mathcal{A}(k) = \mathcal{A}(\Lambda, k)$. Suppose (4) is true for all $k \leq n$. Then for any $i, j \leq n$ with $i \leq j \leq i + 1$ and for any $B \in \mathcal{A}(i),\ C \in \mathcal{A}(j)$ with $B \cup C \in \mathcal{A}(i + j)$, we have

$$\begin{aligned} P_\Lambda(X_{B \cup C} = 1) \quad \geq \quad & \rho \cdot \sqrt{P_\Lambda(X_B = 1) \cdot P_\Lambda(X_B = 0) \cdot P_\Lambda(X_C = 1) \cdot P_\Lambda(X_C = 0)} \\ & + P_\Lambda(X_B = 1) \cdot P_\Lambda(X_C = 1) \end{aligned}$$

Define $\alpha = \log_2 i$ and $\beta = \log_2 j$. Since $P_\Lambda(X_B = 1) \leq \frac{1}{2}$ and $P_\Lambda(X_C = 1) \leq \frac{1}{2}$, and $x \mapsto x(1 - x)$ is increasing on $[0, \frac{1}{2}]$:

$$\begin{aligned} P_\Lambda(X_{B \cup C} = 1) \quad & \geq \quad \rho \cdot \sqrt{\xi \cdot \rho^\alpha (1 - \xi \cdot \rho^\alpha) \cdot \xi \cdot \rho^\beta (1 - \xi \cdot \rho^\beta)} + \xi \cdot \rho^\alpha \cdot \xi \cdot \rho^\beta \\ & \geq \quad \xi \cdot \rho^{\frac{\alpha+\beta}{2}+1} \cdot \sqrt{(1 - \xi \cdot \rho^\alpha) \cdot (1 - \xi \cdot \rho^\beta)} \ + \ \xi^2 \cdot \rho^{\alpha+\beta} \end{aligned}$$

Since $\beta \geq \alpha$, we have $1 - \xi \rho^\beta \geq 1 - \xi \rho^\alpha$ and hence:

$$\begin{aligned} P_\Lambda(X_{B \cup C} = 1) \quad & \geq \quad \xi \cdot \rho^{\frac{\alpha+\beta}{2}+1} \cdot \sqrt{(1 - \xi \cdot \rho^\alpha) \cdot (1 - \xi \cdot \rho^\alpha)} \ + \ \xi^2 \cdot \rho^{\alpha+\beta} \\ & \geq \quad \xi \cdot \rho^{\frac{\alpha+\beta}{2}+1} \cdot (1 - \xi \cdot \rho^\alpha) \ + \ \xi^2 \cdot \rho^{\alpha+\beta} \\ & = \quad \xi \cdot \rho^{\frac{\alpha+\beta}{2}+1} \cdot \left(1 - \xi \cdot \rho^\alpha + \xi \cdot \rho^{\frac{\alpha+\beta}{2}-1}\right) \\ & \geq \quad \xi \cdot \rho^{\frac{\alpha+\beta}{2}+1} \cdot \left(1 + \xi \cdot \left(\rho^{\frac{\alpha+\beta}{2}-1} - \rho^\alpha\right)\right) \end{aligned}$$

Now $i \geq 1$, $j \leq i + 1$ implies $j \leq 4i$ and hence $\log_2 j \leq \log_2 i + 2$. It follows that $\beta \leq \alpha + 2$ and $\rho^{\frac{\alpha+\beta}{2}-1} \geq \rho^\alpha$. As a result,

$$P_\Lambda(X_{B \cup C} = 1) \quad \geq \quad \xi \cdot \rho^{\frac{\alpha+\beta}{2}+1}$$

By the concavity of $u \to \log_2 u$:

$$\frac{\log_2 i + \log_2 j}{2} + 1 \leq \log_2\left(\frac{i+j}{2}\right) + 1 \leq \log_2(i+j),$$

and therefore

$$P_\Lambda(X_{B\cup C} = 1) \quad \geq \quad \xi \cdot \rho^{\log_2(i+j)}$$

To conclude the proof, if (4) is true for every $k < n$, and if $A \in \mathcal{A}(n+1)$, then if $n+1$ is even (respectively, odd), $\exists B \in \mathcal{A}(\frac{n+1}{2})$, $C \in \mathcal{A}(\frac{n+1}{2})$ (respectively, $\exists B \in \mathcal{A}(\frac{n}{2})$, $C \in \mathcal{A}(\frac{n}{2}+1)$), with $A = B \cup C$ and $\rho(B,\ C) \geq \rho$. Hence, $P_\Lambda(X_A = 1) = P_\Lambda(X_{B\cup C} = 1) \geq \xi \cdot \rho^{\log_2(n+1)}$.

Appendix B: Error Rates

We justify the statement that our detector F minimizes the false positive error rate among all false negative zero detectors. To simplify matters, let us suppose that $P(I) > 0$ for every $I \in \mathcal{I}$; it follows that $P_\Lambda(I) > 0$ for every $I \in \mathcal{I}_\Lambda$, the set of images containing an object with pose in Λ. Let $f : \mathcal{I} \longrightarrow \{0,1\}$ be any detector and recall that $\alpha(f)$ is the false negative error $P_\Lambda(f=0)$. Then $\alpha(f) = 0$ if and only if $\mathcal{I}_F \subset \{f = 1\}$. In particular, the condition $\Gamma \subset \{f = 1\}$ implies $\alpha(f) = 0$ because $I \in \Lambda' \subset \Lambda$ implies that $f_\Lambda(I) = 1$ (since f_Λ is an invariant test for Λ) and hence $\mathcal{I}_F \subset \Gamma$.

Suppose f depends on I only through the family of tests $\{f_{m,l}(I)\}$. Suppose further that every possible set of test values $\{f_{m,l}(I)\} \in \{0,1\}^{\sum L_m}$ consistent with $I \in \Gamma$ is realized by some object image $I \in \mathcal{I}_F$. Then the condition $\Gamma \subset \{f = 1\}$ is also necessary for $\alpha(f) = 0$. In other words, f has zero false negative error if and only if $f(I) = 1\ \forall I \in \Gamma$. Consequently, the smallest false positive error is achieved by setting $f(I) = 1$ if and only if $I \in \Gamma$, i.e., choosing $f = F$.

Appendix C: Mean Computation

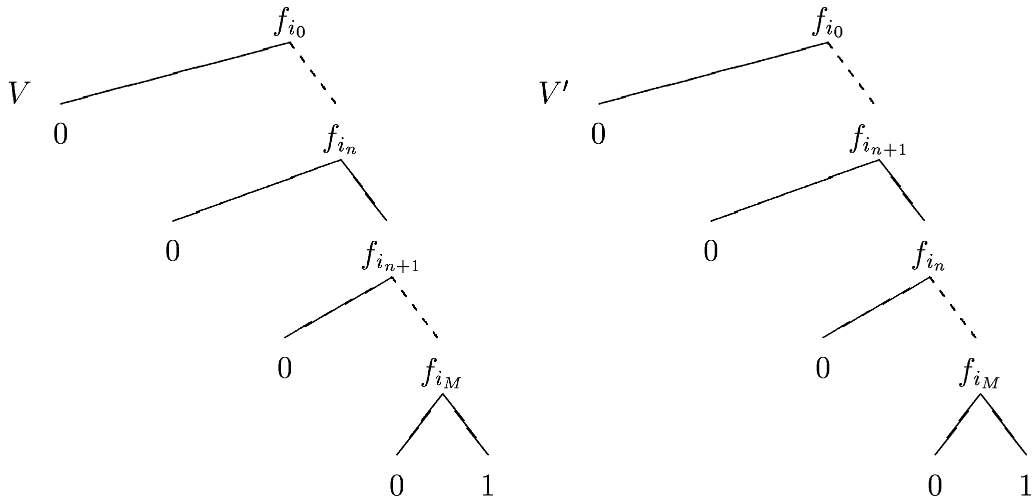


Figure 20: The vine V' is a rearrangement of V which has lower cost if $i_{n+1} < i_n$.

Consider first detecting a target, represented by a *single conjunction of attributes*, versus a background hypothesis which is a priori far more likely. For example, we must separate Napoleon from all other prominent historical figures. Let $f_0, ..., f_M$ be the binary random variables corresponding to the attributes; thus the target is represented by $\prod_m \{f_m = 1\}$. We test sequentially. Background is declared upon the first negative test and hence all the tests are eventually performed when the target is present. This procedure is represented by the labeled vine V in Figure 20 where i_m is the index of the m'th test performed.

Clearly all such procedures have no false negative error and the minimum possible false positive error based on the given attributes. We therefore seek the least expensive V in terms of mean computation. Since the background hypothesis is assumed dominant, the mean is computed relative to P_0. Suppose the tests are independent under P_0, with

$$P_0(f_m = 0) = \beta_m, \ \ m = 0, ..., M.$$

Thus $1 - \beta_m$ the incidence in the background population. We can suppose (by relabeling the attributes) that

$$0 < \beta_0 \leq \beta_1 \leq \cdots \leq \beta_M < 1. \tag{7}$$

Let $c_0, ..., c_M$ denote the costs. The cost of V, denoted $C(V)$, is the sum of the costs of the tests performed before reaching a terminal node, and hence a random variable. The mean cost can be computed by summing, over all internal nodes t of V, the cost of the test at t times the probability of reaching t, yielding:

$$E_0(C(V)) = c_{i_0} + \sum_{m=1}^{M} c_{i_m} \prod_{l=1}^{m-1} (1 - \beta_{i_l}).$$

If $c_m \equiv 1$, the mean cost is simply the average number of tests performed. The best procedure is then $i_m = M - m$, which proceeds from rare to common. In this case the false positive error is clearly $\prod_{m=0}^{M}(1 - \beta_m)$. Notice that under the independence assumption, a background instance can land in the all "1" leaf of the vine together with the object.

However, equal costs is not realistic. General tests (common attributes) should be inexpensive to test whereas dedicated tests (rare attributes) should be costly. For instance, if the cost behaves like an (approximate) code length, then $c_m \approx -\log_2(1 - \beta_m)$. Suppose, in fact, we assume that $c_m = \Phi(\beta_m)$, where

$$\Phi : [0,1] \to [0,1], \;\; \Phi(0) = 0,$$

and Φ is strictly increasing and convex.

Proposition: *Under the above cost structure, the best strategy for detecting a single conjunction of attributes is $i_m = m$, which is coarse-to-fine in likelihood.*

Example: The best procedure to check for Napoleon is then *deceased?* → *general?* → *Corsican?*

Proof: Let V denote the vine in Figure 20. Suppose V is optimal but that $i_m \neq m$

for some m. Then $i_{n+1} < i_n$ for some n. The mean cost of V is

$$\begin{aligned} E_0(C(V)) &= c_{i_0} + \sum_{m=1}^{n-1} c_{i_m} \prod_{l=1}^{m-1} (1-\beta_{i_l}) \\ &+ c_{i_n} \left((1-\beta_{i_1})\cdots(1-\beta_{i_{n-1}})\right) + c_{i_{n+1}} ((1-\beta_{i_1})\cdots(1-\beta_{i_n})) \\ &+ \sum_{m=n+2}^{M} c_{i_m} \prod_{l=1}^{m-1} (1-\beta_{i_l}) \end{aligned}$$

Let V' be the same vine as V, but with the positions of f_{i_n} and $f_{i_{n+1}}$ reversed, as in Figure 20. The mean cost of V' has a similar expression, with the same first and last terms, but with the middle term replaced by

$$c_{i_{n+1}} \left((1-\beta_{i_1})\cdots(1-\beta_{i_{n-1}})\right) + c_{i_n} \left((1-\beta_{i_1})\cdots(1-\beta_{i_{n-1}})(1-\beta_{i_{n+1}})\right).$$

Therefore

$$\begin{aligned} E_0(C(V)) - E_0(C(V')) &= c_{i_n} \left((1-\beta_{i_1})\cdots(1-\beta_{i_{n-1}})\right) + c_{i_{n+1}} ((1-\beta_{i_1})\cdots(1-\beta_{i_n})) \\ &- c_{i_{n+1}} \left((1-\beta_{i_1})\cdots(1-\beta_{i_{n-1}})\right) \\ &- c_{i_n} \left((1-\beta_{i_1})\cdots(1-\beta_{i_{n-1}})(1-\beta_{i_{n+1}})\right) \\ &= \left(c_{i_n}\beta_{i_{n+1}} - c_{i_{n+1}}\beta_{i_n}\right) \prod_{l=1}^{n-1}(1-\beta_{i_l}) \\ &> 0. \end{aligned}$$

The last inequality results from convexity and contradicts optimality. Hence $i_m = m$ for all m.

Finally, consider a corresponding model for a *disjunction of conjunctions*, and the corresponding optimality of T^* among all binary trees in $\mathcal{T}$ which represent f. As for the cost structure, for $T \in \mathcal{T}$, let B_t denote the event of reaching node t. The cost $C(T)$ of $T \in \mathcal{T}$ is

$$C(T) = \sum_t I_{B_t} C_t$$

where the sum is over all leaves of T and C_t is the sum of the costs along the branch from the root to t. The mean cost is

$$E_0(C(T)) = \sum_t P_0(B_t)C_t = \sum_s P_0(B_s)c_{m_s}$$

where the second sum is over all *internal* nodes of T and the test at node s is (m_s, l_s).

The hypotheses $\mathcal{H}$ in Theorem 2 refer to the following three assumptions:

- *The tests are conditionally independent under* P_0.
- *The distribution of* $f_{m,l}$ *depends only on* m, *with* $\beta_m = P_0(f_{m,l} = 0)$ *and the ordering in (7).*
- *The cost of* $f_{m,l}$ *depends only on* m, *with* $c_m = \Phi(\beta_m)$ *and* Φ *as above.*

Notice that (7) is now a genuine assumption.

References

Amit, Y. (1999), 'A neural network architecture for visual selection', *Neural Computation* .

Amit, Y. & Geman, D. (1997), 'Shape quantization and recognition with randomized trees', *Neural Computation* **9**, 1545–1588.

Amit, Y. & Geman, D. (1999), 'A computational model for visual selection', *Neural Computation* **11**, 1691–1715.

Baum, E. B. & Haussler, D. (1989), 'What size net gives valid generalization?', *Neural Comp.* **1**, 151–160.

Cootes, T. F. & Taylor, C. J. (1996), Locating faces using statistical feature detectors, *in* 'Proceedings, Second International Conference on Automatic Face and Gesture Recognition', IEEE Computer Society Press, pp. 204–209.

Devroye, L., Gyorfi, L. & Lugosi, G. (1995), *Probabilistic methods for pattern recognition*, Springer-Verlag, Berlin.

Fleuret, F. (2000), Dtection hirarchique de visages par apprentissage statistique, PhD thesis, University of Paris VI, Jussieu, France.

Geman, D. & Jedynak, B. (1996), 'An active testing model for tracking roads from satellite images', *IEEE Trans. PAMI* **18**, 1–15.

Grimson, W. E. L. (1990), *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, Massachusetts.

Haiyuan, W., Qian, C. & Masahiko, Y. (1999), 'Face detection from color images using a fuzzy pattern matching method', *IEEE Trans. PAMI* **10**.

Jedynak, B. & Fleuret, F. (1996), Reconnaissance d'objets 3d l'aide d'arbres de classification, *in* 'Proc. Image'Com 96', Bordeaux, France.

Lamdan, Y., Schwartz, J. T. & Wolfson, H. J. (1988), Object recognition by affine invariant matching, *in* 'Proc. IEEE Conf. on Computer Vision and Pattern Recognition', pp. 335–344.

Leung, T., Burl, M. & Perona, P. (1995), Finding faces in cluttered scenes using labeled random graph matching, *in* 'Proceedings, 5th Int. Conf. on Comp. Vision', pp. 637–644.

Maurer, T. & von der Malsburg, C. (1996), Tracking and learning graphs and pose on image sequences of faces, *in* 'Proceedings, Second International Conference on Automatic Face and Gesture Recognition', IEEE Computer Society Press, pp. 176–181.

Miao, J., Yin, B., Wang, K., Shen, L. & Chen, X. (1999), 'A hierarchical multiscale and multiangle system for human face detection in complex background using gravity-center template', *Pattern Recognition* **32**, 1237–1248.

Ming, X. & Akatsuka, T. (1998), Multi-module method for detection of a human face from complex backgrounds, *in* 'Proceedings of the SPIE', pp. 793–802.

Osuna, E., Freund, R. & Girosi, F. (1997), Training support vector machines: an application to face detection, *in* 'Proceedings, CVPR', IEEE Computer Society Press, pp. 130–136.

Rojer, A. S. & Schwartz, E. L. (1992), A quotient space hough transform for space variant visual attention, *in* G. A. Carpenter & S. Grossberg, eds, 'Neural Networks for Vision and Image Processing', MIT Press.

Rowley, A. R. (1999), Neural Network-Based Face Detection, PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Rowley, H. A., Baluja, S. & Kanade, T. (1998), 'Neural network-based face detection', *IEEE Trans. PAMI* **20**, 23–38.

Sabert, E. & Tekalp, A. M. (1998), 'Frontal-view face detection and facial feature extraction using color, shape, and symmetry-based cost functions', *IEEE Trans. PAMI* **19**, 669–680.

Sung, K. K. & Poggio, T. (1998), 'Example-based learning for view-based face detection', *IEEE Trans. PAMI* **20**, 39–51.

Ullman, S. (1996), *High-Level Vision*, M.I.T. Press, Cambridge, MA.

Vapnik, V. (1996), *The Nature of Statistical Learning*, Springer-Verlag, Berlin.

Wee, S., Ji, S., Yoon, C. & Park, M. (1998), Face detection using pattern information and deformable template in motion images, *in* 'Proc. Fifth Inter. Conf. on Soft Computing and Information/Intelligent Systems', pp. 213–216.

Wilder, K. (1998), Decision tree algorithms for handwritten digit recognition, PhD thesis, University of Massachusetts, Amherst, Massachusetts.

Yuille, A. L., Cohen, D. S. & Hallinan, P. (1992), ‘Feature extraction from faces using deformable templates’, *Inter. J. Comp. Vision* **8**, 104–109.

Zhu, S. C., Wu, Z. N. & Mumford, D. (1997), ‘Minimax entropy principle and its application to texture modeling’, *Neural Computation* **9**, 1627–1660.