The Trace Model for Object Detection and Tracking

Sachin Gangaputra[1] and Donald Geman[2]

[1] Dept. of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore, MD 21218.
sachin@jhu.edu
[2] Dept. of Applied Mathematics and Statistics
The Johns Hopkins University
Baltimore, MD 21218.
geman@jhu.edu

Abstract. We introduce a stochastic model to characterize the online computational process of an object recognition system based on a hierarchy of classifiers. The model is a graphical network for the conditional distribution, under both object and background hypotheses, of the classifiers which are executed during a coarse-to-fine search. A likelihood is then assigned to each history or "trace" of processing. In this way, likelihood ratios provide a measure of confidence for each candidate detection, which markedly improves the selectivity of hierarchical search, as illustrated by pruning many false positives in a face detection experiment. This also leads to a united framework for object detection and tracking. Experiments in tracking faces in image sequences demonstrate invariance to large face movements, partial occlusions, changes in illumination and varying numbers of faces.

1 Introduction

The two main categories of traditional pattern classification methods are generative and discriminative [1]. Generative methods involve the design and estimation of a probability distribution over features, both observed and unobserved, which capture the appearance of patterns in each class. Recent variations include work on spatial arrangements of parts [2], Boolean models [3], reusable parts [4] and compositional vision [5]. Such methods usually require intense computation (e.g., computing MAP estimators) and extensive modeling. Nonetheless, in principle, they can account for context and semantic labels at many levels, thereby providing a comprehensive analysis of natural scenes. In contrast, discriminative methods usually aim at inducing decision surfaces directly from training data. Popular methods include support vector machines [6, 7], neural networks [8] and Adaboost [9]. Some are well-grounded in the theory of inductive learning and achieve high performance in classification. However, they often require very large training sets and their extension to global, full-scale scene interpretation is

by no means obvious. Of course, these categories are hardly disjoint and many methods, including those proposed here, involve components of both.

Recently, a different approach has been applied to pattern recognition [10–12]. In *computational modeling*, the primary object of analysis is the online computational process rather than probability distributions or decision surfaces. Hierarchies of binary classifiers which cover varying subsets of hypotheses are built from standard discriminative methods by exploiting shared properties of the appearance of shapes. Online, the hierarchy is traversed using a coarse-to-fine (CTF) search strategy: a classifier is evaluated if and only if all its ancestors have been evaluated and were positive. One important consequence is that computation is concentrated on ambiguous regions of the image; in particular, the "object hypothesis" is rejected as quickly as possible in background regions. A limitation of this approach is that statistical interactions among the classifiers in the hierarchy is not taken into account; in particular, no global likelihoods are assigned.

Here, we extend computational modeling, and take a step towards contextual analysis, by introducing a global stochastic network to model classifier interactions. The central concept is the *trace* of processing, which encodes the *computational history* – the family of classifiers performed, together with their outcomes, during CTF search. Notice that the trace is a far richer structure than the output of a decision tree; it is in fact a data-driven subtree of the original hierarchy since many branches may be partially traversed before a negative result is encountered or a leaf is reached. The trace space is represented by a tree-structured graphical network and a likelihood is assigned to each trace under both object and background hypotheses. This provides a generative framework for the hierarchy of classifiers. Detections (full chains of positive responses) can then be analyzed using likelihood ratio tests, adding a statistical component to sequential search strategies.

We test the effectiveness of our trace model in experiments in face detection and face tracking. Single-frame detection is based on the CTF framework proposed in [11], where a hierarchy of linear classifiers is used to efficiently reject non-face patterns and focus computation on face-like regions. Likelihood ratios of observed traces represent a measure of confidence for each detection, allowing for higher discrimination than with purely CTF search. This is illustrated by successfully pruning false positives to produce a strictly superior ROC curve. Tracking of faces in a video sequence is accomplished by integrating frame-based probability measures within a spatial-temporal Markov model for the joint evolution of poses and traces. Due to continuously updating detections, there are no restrictions on face movements. Unlike existing approaches, the motion model is not used to restrict the search domain but rather only to link detections between consecutive frames. This framework then unites detection and tracking within a single stochastic model.

In Section 2, we provides an overview of hierarchical object detection. The trace model is introduced in Section 3, along with the construction of a probability distribution on the space of traces relative to a general hierarchy of classifiers.

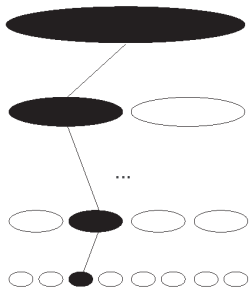

Fig. 1. Hierarchical class/pose decomposition. Each cell represents a subset of classes and poses. An alarm is identified with a fine (leaf) cell Λ if the classifiers for every coarser cell (i.e containing Λ) responds positively.

In Section 4, we specialize to the case of learning a trace model for a hierarchy based on the pose of a frontal view of a face and in Section 5 we demonstrate how this model can eliminate false positives in hierarchical face detection. In Section 6, the spatial trace model is integrated into a spatial-temporal Markov model in order to produce a real-time face tracking system. Concluding remarks are provided in Section 7.

2 Hierarchical Object Detection

Object detection refers to discovering and localizing instances from a list of object classes based on a grey level image of an underlying scene. The basic hierarchical framework can be found in [10–13]. In hierarchical detection, both learning and parsing algorithms are based on a tree-structured representation of hypotheses – a sequence of nested partitions – which captures shared structure, e.g., common shape features. Hypotheses correspond to individual class/pose pairings, although the framework is more general. Whereas scene interpretations may involve multiple, inter-connected pairings, we shall focus on pure detection. Each cell of the hierarchy corresponds to a subset of hypotheses and is included in exactly one of the cells in the preceding, coarser partition (see Fig.1). Fine cells may not correspond to individual hypotheses.

A binary classifier X_η is associated with the cell at each $\eta \in T$, where T denotes the tree graph underlying the hierarchy. Classifier X_η is designed to respond positively ($X_\eta = 1$) to all images labeled by the cell at η and negatively ($X_\eta = -1$) to as many images as possible which fall into a suitable alternative category. These classifiers range from those near the root of T, which accommodate many hypotheses simultaneously, to those near the leaves of T, which are more dedicated (and hence selective). In principle, the classifiers could be constructed by any learning algorithm, but under the constraint that each classifier maintain a very small false negative error rate, which facilitates early termination of the search.

Scenes are parsed by a coarse-to-fine exploration of the hierarchy, i.e., starting at the root and evaluating a classifier if and only if all ancestors have been evaluated and returned a positive answer. This processing strategy is known to be theoretically optimal [10] under certain assumptions about how the power and cost of the classifiers are related and how these quantities interact with the "scope" of the classifiers – the number of hypotheses covered.

The result of processing an image is then the list of hypotheses determined by the union of all leaf cells $\eta \in \partial T$ with the property that $X_\eta = 1$ and all classifiers at ancestors of η also respond positively. This can be visualized as a *chain of positive responses* in the hierarchy of cells (see Fig. 1). Areas of the image rejected by coarse tests are then rapidly processed, whereas ambiguous areas are not labeled until at least some fine classifiers have been evaluated. The resulting distribution of processing is highly skewed and detection is rapid at the expense of some false positives.

On an empirical level, the success of this technique has been demonstrated in several contexts, including experiments in face detection [11, 13] and multi-class character recognition [12]. In the former case, for example, parsing an image results in a binary decision labeling each non-overlapping $k \times k$ window (e.g., $k = 16$) as either "background" or "face". Although this method is quite fast and accurate (see Section 5 for comparisons with other methods), it does not assign any numeric confidence measure to each detection, which can aid in resolving competing interpretations. More generally, there is no global stochastic model for the hierarchy of classifiers.

The key to introducing a model, and accounting for context, is to exploit the rich information provided by hierarchical search. Information is lost by only collecting the list of complete chains. Clues about the semantic identity of image regions, specifically the existence and presentations of objects of interest, can be accumulated by considering the global history of the search process. More specifically, processing a subimage leaves a fairly distinctive "signature" because *every test tells us something about every possible interpretation.* That is, if $y \in Y$ is an interpretation (e.g., face at some pose), then each classifier X_η in the hierarchy offers some evidence for the presence or absence of y, *even if* X_η *is based on a subset* $\Lambda_\eta \subset Y$ *which does not contain* y. The trace model is intended to capture this type of global information.

3 The Trace of Coarse-to-Fine Search

Our approach is to model the computational history using a graphical stochastic network indexed by certain subtrees of T. This then provides a joint probability distribution over all possible processing records. The nature of the coarse-to-fine processing makes this feasible as the search imposes major restrictions on the possible records – subtrees – that can be observed. This in turn leads to a simple distribution on the search histories or "traces" and provides a natural likelihood-ratio test for weeding out false detections.

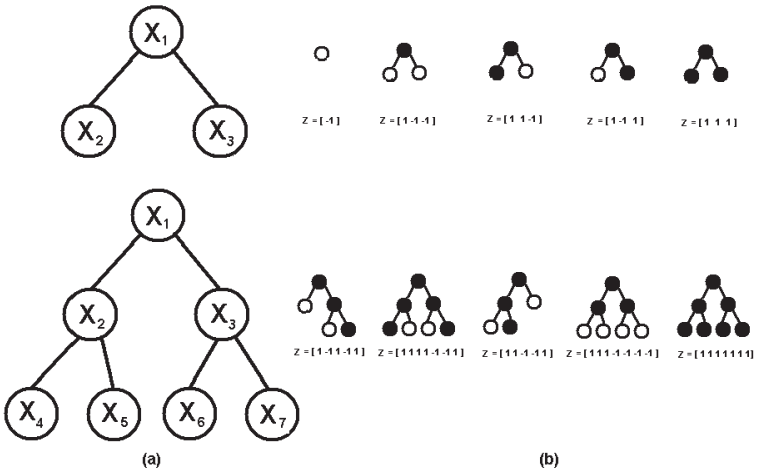

Fig. 2. The result of CTF search is a labeled subtree where dark circles indicate a positive classifier result and light circles a negative result. The traces are depicted together with the outcomes of the classifiers performed. Top panel: (a) A hierarchy of three classifiers; (b) The 5 different traces that can result from CTF search. Bottom panel: (a) A hierarchy of seven classifiers; (b) Five of the 26 possible traces for this hierarchy.

3.1 Trace Configurations

Depending on the image I, certain nodes $\eta \in T$ are visited during CTF search and their corresponding classifiers $X_\eta \in \{-1, 1\}$ are evaluated. The result of CTF search is then a *labeled subtree* of T, which we call the *trace* of image I. The nodes of the trace correspond to the classifiers evaluated and the labels correspond to the outcomes. Specifically, let $S(I) \subset T$ denote the set of visited nodes, a *random subtree*, and write $Z(I) = \{X_\eta, \eta \in S(I)\} \in \mathcal{Z}$ for the trace, where $\mathcal{Z}$ denotes the set of all possible traces for a given hierarchy. In addition, let $\mathcal{A}_\eta$ denote the set of parent nodes of η. For any trace $Z(I)$, certain constraints result from the fact that a classifier X_η is performed if and only if all ancestor classifiers $\{X_\xi, \xi \in \mathcal{A}_\eta\}$ are performed *and* each one is positive. In particular, i) the classifier at any non-terminal node of $S(I)$ must be positive; ii) the classifier at any node which is terminal in $S(I)$ but not terminal in T must be negative; and iii) the classifier at a terminal node of both $S(I)$ and T can be either positive or negative.

The situation is illustrated in Fig. 2 for two simple binary hierarchies. For three nodes and three corresponding binary classifiers X_1, X_2, X_3, there are 2^3 total possible full realizations but only five possible traces, listed in the upper right of Fig. 2. With seven nodes and classifiers, there are $2^7 = 128$ full realizations and twenty-six possible traces, five of which are shown in the lower right of Fig. 2. In general, the total number of traces depends on T.

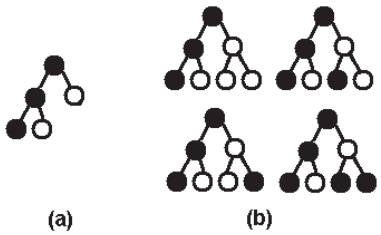

Fig. 3. (a) A trace Z from a binary hierarchy with three levels; (b) All possible full configurations $\mathbf{X}$ that could result in Z.

The hierarchies we construct in our experiments are not binary. However, in the binary case, there is a simple relationship between the number of subtrees, $n_{sub}(k)$, of a tree T with depth k (the depth of the root is 1) and the number of traces, $n_{tr}(k) = |\mathcal{Z}|$. In fact, it is easy to show there is a one-to-one correspondence between $\mathcal{Z}$ and the subtrees of a tree of depth $k+1$, and hence $n_{tr}(k) = n_{sub}(k+1)$. Given the trace of a tree of depth k, expanding every "on" node (which is necessarily terminal) into two children gives a subtree of a tree of depth $k+1$; conversely, every subtree can be identified with a trace by cutting off its terminal leaves. It follows that

$$n_{tr}(k) = n_{sub}(k+1) = n_{sub}^2(k) + 1 = n_{tr}^2(k-1) + 1, \ k \geq 2.$$

In particular, $n_{tr}(1) = 2$, $n_{tr}(2) = 5$ and $n_{tr}(3) = 26$ (see Fig. 2(b)).

3.2 Trace Distributions

CTF search induces a mapping $\tau : \{-1, 1\}^T \rightarrow \mathcal{Z}$ from full configurations $\mathbf{X}$ to traces Z. In general, many realizations $\mathbf{X}$ are mapped to the same trace Z. In Fig. 3, the four configurations in (b) are mapped to the trace in (a). This mapping induces a partition of the entire configuration space. Consequently, given any probability distribution $p_{\mathbf{X}}$ for $\mathbf{X}$, we have

$$\sum_{z \in \mathcal{Z}} p_{\mathbf{X}}(\tau^{-1}(z)) = 1. \tag{1}$$

However, in order to construct a distribution on $\mathcal{Z}$ we need not start with a distribution on the full configuration space.

One natural distribution on $\mathcal{Z}$ can be constructed directly along the lines of graphical models. This direct construction has the added advantage that the model requires only one parameter for each node in T. In contrast, learning a graphical model for $\mathbf{X}$ on the full realization space $\{-1, 1\}^T$ can be difficult for large T even with conditional independence assumptions since the number of nodes, as well as the number of parameters determining each conditional probability, increases exponentially with $|T|$. Moreover, in terms of online computation, the original motivation for constructing a hierarchy of classifiers under

the zero false negative constraint was the amount of computation involved in evaluating classifiers at many image locations and resolutions.

Theorem 1. *Let* $\{p_\eta,\ \eta \in T\}$ *be any set of numbers with* $0 \le p_\eta \le 1$. *Then*

$$P(z) = \prod_{\eta \in S_z} p_\eta(x_\eta) \tag{2}$$

defines a probability distribution on traces where S_z *is the subtree identified with* z *and* $p_\eta(1) = p_\eta$ *and* $p_\eta(-1) = 1 - p_\eta$.

Proof. There are several ways to prove that (2) implies $\sum_z P(z) = 1$ using the type of "peeling" argument common in graphical models. The "direct" proof proceeds by performing the summation one node at a time starting from the leaves of T. Start with any terminal node η of T and divide all traces into three disjoint groups: those for which S does not contain η; those for which $\eta \in S$ and $x_\eta = 1$; and those for which $\eta \in S$ and $x_\eta = -1$. The second and third groups are of equal size and there is a one-to-one pairing between them in which each pair is the same trace except for the sign of x_η. Adding the probabilities in each pair, and using $p_\eta(1) + p_\eta(-1) = 1$, results in a reformulation of the problem with probabilities identical to those in (2) except that node η does not appear, i.e., the trace space is relative to $T \setminus \{\eta\}$. Recursively looping over all the leaves of T then reduces the problem to a hierarchy of depth $k-1$; continuing this way, proving $\sum_z P(z) = 1$ eventually reduces to $p_\eta(1) + p_\eta(-1) = 1$ for the root η of T.

□

There is obviously a natural connection between the trace distribution given by (2) and a graphical model $p_{\mathbf{X}}$ on the full configuration space, linked by defining

$$p_\eta(x_\eta) = p_{\mathbf{X}}(x_\eta | x_\xi = 1, \xi \in \mathcal{A}_\eta). \tag{3}$$

Here, the distribution $p_{\mathbf{X}}$ on $\{-1, 1\}^T$ is determined by imposing the splitting property of DAGs [14]:

$$p_{\mathbf{X}}(\mathbf{x}) = P(X_\eta = x_\eta, \eta \in T) = \prod_{\eta \in T} P(X_\eta = x_\eta | X_\xi = x_\xi, \xi \in \mathcal{A}_\eta). \tag{4}$$

We can choose any graphical model $p_{\mathbf{X}}$ consistent with (3). Then we only need to show that (2) holds; normalization is guaranteed by the mapping from full realizations to traces. Proving this is again a standard argument in graphical models. In fact, (3) holds relative to *any* sub-configuration on a subtree of T (i.e., whether or not the node histories consist of all positive responses). In particular, if $\Omega(z)$ is the subset of the full configuration space that maps to trace z, we clearly have:

$$\begin{aligned} P(Z = z) &= \sum_{\mathbf{x} \in \Omega(z)} p_{\mathbf{X}}(\mathbf{x}) \\ &= \sum_{\mathbf{x} \in \Omega(z)} \prod_{\eta \in T} p(x_\eta | x_\xi, \xi \in \mathcal{A}_\eta). \end{aligned}$$

This reduces to (2) by factoring the product of conditional probabilities into two groups and by extracting common terms in a recursive fashion.

The important point is that the conditional probabilities in the full model are reduced to binomial terms $p_\eta(x_\eta)$ since all the conditional events are "positive histories." *Consequently, specifying a single parameter* $p_\eta(1)$ *for every node* $\eta \in T$ *yields a consistent probability model on traces.* In contrast, in the full model with binary trees, 2^k parameters would be required to specify each conditional probability for a history of length k, and hence order 4^k parameters would be required altogether, at least without imposing further Markov assumptions on path histories.

4 Learning Trace Models for a Pose Hierarchy

In this section, we specialize the trace formulation to the case of a pose hierarchy for faces. A reference set of poses is recursively partitioned into finer and finer cells Λ_η and the classifier X_η for cell η is designed to detect all faces with poses in Λ_η. The manner in which the classifiers are constructed from training data, and full scenes are processed, will be reviewed only briefly in the following section since these issues have been discussed in previous work; for example further details may be found in [11] and [12]. Here we review what the hierarchy represents in order to understand what the corresponding trace distributions mean and how they are estimated from data.

4.1 Pose Hierarchy

The space of hypotheses is the set of poses of a face. Each classifier is trained on a specific subset of face subimages which satisfy certain pose restrictions. In detecting frontal views of faces, tilts are restricted to the range $-15^\circ \leq \alpha \leq 15^\circ$. The base detector is designed to detect faces with scales (the number of pixels between the eyes) in the range $8 \leq s \leq 16$. The position of the face (taken to be the midpoint between the eyes) is unrestricted. To detect larger faces, the original image is downsampled before applying the base detector. With four levels of downsampling, one is able to detect faces with sizes from 8 to 128 pixels.

Processing an entire image with a single hierarchy of classifiers would entail building a root classifier which applies to *all* face positions simultaneously, and to tilts and scales in the ranges given above. Instead, the face location in the coarsest cell in the hierarchy is restricted to an 8×8 block and the entire image is processed by visiting each (non-overlapping) 8×8 block and applying the base detector to the surrounding image data. Specifically, then, the classifier at the root of the hierarchy is designed to detect faces with tilts in the range $-15^\circ \leq \alpha \leq 15^\circ$, scales in the range $8 \leq s \leq 16$, and location restricted to an 8×8 window. The leaf cells localize faces to a 2×2 region with $\Delta\alpha = 10^\circ$ and $\Delta s = 2$ pixels. In particular, faces are not detected at the resolution of one specific position, scale and tilt, but rather at the resolution of the leaf cells. For

ease of notation, however, each leaf cell $s \in \partial T$ in the hierarchy T is represented by a single pose in that cell, call it θ_s.

The discussion in Section 3 about constructing trace distributions can now be applied conditionally on each leaf cell s, i.e., under the hypothesis that there is a face with pose in Λ_s. Using the representative pose θ_s to signify this hypothesis, the conditional probability of observing a trace z in the pose hierarchy is then

$$P(z|\theta_s) = \prod_{\eta \in S_z} p_\eta(x_\eta|\theta_s). \tag{5}$$

4.2 Learning

The task of learning is then to estimate the probabilities $p_\eta(1|\theta_s), \eta \in T$, for each leaf cell $s \in \partial T$. Recall that this probability represents the likelihood that the classifier at node η responds positively *given that all its ancestors have responded positively*. In addition, detection will involve a likelihood ratio test for the hypothesis "θ_s" against a universal "background hypothesis," denoted by B. Under B, the trace data follow another distribution estimated from non-face subimages. Consequently, we must also learn the probabilities $p_\eta(1|B), \eta \in T$.

Due to the natural assumption of space-invariance (i.e., the trace distributions are block-independent), we need to only learn the responses of classifiers for all poses contained within a single, reference block. Moreover, two pose cells at the same level in the (reference) hierarchy which differ only in the location of the subset of positions (i.e., cover the same subset of scales and tilts and the same subset of positions up to translation) can evidently be aggregated in collecting statistics. Notice also that, in estimating $p_\eta(1|\theta_s)$ for a fixed η, all the face training data with poses in the leaf cell represented by θ_s are also aggregated in compiling empirical statistics.

The model parameters are learned for the object model by accumulating the results of classification tests over a standard face database and for the background model from subimages randomly sampled from the WWW. Fig. 4 illustrates the distribution of the model parameters $p_\eta(1|\theta_s)$ for one specific pose θ_s and under the background hypothesis. Only the section of the hierarchy that contains the complete chain corresponding to the pose θ_s is illustrated. A darker circle indicates a higher value of the probability $p_\eta(1|\theta_s)$. As expected, we observe darker circles along the chain that corresponds to the true pose. A consistent decrease in the darkness at deeper levels is observed for the background model.

5 Experiments in Face Detection

We now demonstrate the advantage of the trace model with respect to the baseline detector utilized in previous work [12, 10, 11] on coarse-to-fine object detection. Briefly, the baseline detector operates as follows: The image is partitioned into disjoint 8×8 blocks and the image data surrounding each block is processed by the hierarchy of classifiers which corresponds to the (reference) pose hierarchy.

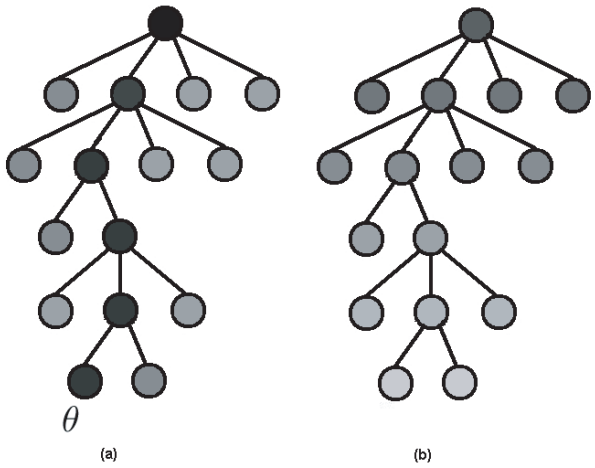

Fig. 4. (a) The learned parameters $p_\eta(1|\theta_s)$ under the hypothesis of a face with pose in the cell represented by θ_s; (b) The same parameters for the background model. Only the section of the full hierarchy that contains the complete chain to the cell represented by θ_s is shown. A circle shaded darker indicates a higher value of the parameter.

The search is breadth-first CTF. A detection is declared at a terminal pose cell ξ when there is a chain in the hierarchy, from the root to ξ, for which $X_\xi = 1$ and $X_\eta = 1, \eta \in \mathcal{A}_\xi$. In the baseline system, when multiple detections are recorded for a given block, some criterion must be used to identify a unique detection for that block. The details are not important for our purposes. The important point is that there is no global probabilistic model for assigning likelihoods to detections or measuring one detection against another, or against a background hypothesis. The trace model provides for this.

The general design of the hierarchy and the classifiers follows previous work [11]. In this work, we use a slightly modified pose hierarchy (see §4.1) and make use of both positive (face) and negative (non-face) training instances in constructing the classifiers. We use the Adaboost [9] learning algorithm to build each $X_\eta, \eta \in T$. The features are the same oriented binary edge fragments from [11]. The same learning algorithm is applied to each cell; only the training set changes. More specifically, a standard training dataset is used to build both the hierarchical classifiers and the trace models. 1600 faces are synthesized from the dataset for each different pose and 10000 randomly selected image patches were downloaded from the WWW and used as "non-faces". The non-face instances used at cell η are those which have responded positively to the preceding classifiers $X_\xi, \xi \in \mathcal{A}_\eta$. In this way, in training X_η, the system is competing with those particular non-faces encountered during CTF search, which increasingly resemble faces.

5.1 Trace-Based Likelihood Ratios

Assume the hierarchy of classifiers has been constructed and processed using the baseline detection system, resulting in a susbset (usually empty) of complete chains for a given 8×8 block W. (Recall that the baseline detector is applied to each in a series of downsampled images in order to detect faces at a wide range of scales.) Let $Z(W)$ denote the trace of block W. For each complete chain in W, say arriving at leaf node $s \in \partial T$, we perform a likelihood ratio test, comparing $P(Z(W)|\theta_s)$ to $P(Z(W)|B)$. A detection is declared "at θ_s" if

$$\frac{P(Z(W)|\theta_s)}{P(Z(W)|B)} \geq \tau.$$

An ROC curve may then be constructed by varying τ and collecting statistics on a test set; see below. The smallest value of τ, i.e., the most conservative in terms of retaining faces at the expense of false positives, is determined by studying the distribution of the likelihood ratio over a training set of faces and non-faces and choosing a value that maintains every face.

NOTES:

- The speed of the algorithm is mainly governed by the baseline detection scheme as the evaluation of each trace likelihood is only performed at complete chains.
- However, restricting the search to complete chains is merely a computational shortcut. Very little would change if screening for complete chains was omitted and hence the likelihood ratio was maximized over *every* pose hypothesis in each block. This is due to the underlying false negative constraint on each classifier. Given a face with pose θ, any trace z which does not contain a path to the leaf containing θ, including a positive value at the leaf, has very small probability compared with $P(z|B)$. As a result, the likelihood ratio is smaller than even the smallest value of τ described above and consequently there is no detection at (the leaf cell containing) θ. *Hence, the detector for block W is effectively a true likelihood ratio test.*

5.2 Towards a Global Model

One might ask whether this block-by-block likelihood ratio test can be related to a full-image search based on a global, generative model. Consider a single hierarchy for the entire image; suppose there is a branch from the root to the subset of poses corresponding to each region W_i, $1 \leq i \leq n$, for a partition of the image pixels into non-overlapping 8×8 blocks. Suppose also that the root test is virtual – always positive. How might the global trace Z be used to make inferences about the poses of all faces in the image? Let Θ denote a collection of poses representing a global (image-wide) hypothesis. Suppose the prior distribution $P(\Theta)$ forbides any two components of Θ with positions in the same 8×8 block W; otherwise it is uniform. (This only rules out severe

occulsion.) We make no assumptions about the number of faces in the image (up to the number of blocks). Let $\Theta = \{\gamma_1, ..., \gamma_n\}$ where $\gamma_i = B$ signals "no face in block W_i" and $\gamma_i = \theta_i$ is a pose with location in W_i.

Let $Z(i)$ correspond to the trace generated with image block W_i. Make the convenient assumptions that that the components of Z are conditionally independent given both Θ and background, that $P(Z(i)|\Theta) = P(Z(i)|\gamma_i)$ and that $P(Z(i)|\gamma_i = B)$ follows a universal "background law" denoted $P(Z|B)$. (The conditional independence assumption is violated in practice because $Z(i)$ depends on the image data surrounding W_i; for example, in the scale range $8 \leq s \leq 16$, faces might occupy a region of order 32×32 and hence adjacent traces have overlapping supports. The other assumptions are reasonable.) Then

$$\frac{P(Z|\Theta)}{P(Z|\mathbf{B})} = \prod_{i=1}^{n} \frac{P(Z(i)|\gamma_i)}{P(Z(i)|B)} = \prod_{i \in F(\Theta)} \frac{P(Z(i)|\theta_i)}{P(Z(i)|B)} \tag{6}$$

where $F(\Theta) \subset \{1, ..., n\}$ is the set of blocks for which $\gamma_i \neq B$. Maximizing this over all Θ is evidently intractable. However, visiting the blocks one-by-one and performing an individual likelihood ratio test is a reasonable approximation.

5.3 Results

Our algorithm is implemented in C++ on a standard Pentium 4 1.8GHz PC, and we use a subset of the CMU+MIT [8, 15] frontal face test set to estimate performance. Images with strong pose variations in 2D and out-of-plane face orientations are removed from the original test set. Figure 5 shows the result of the trace-based system at a high detection rate (i.e., small τ) on a few images of this test set. Processing a 320×240 image takes only a fraction of a second.

Fig. 6 illustrates the difference in detection performance between the trace-based system and the baseline detector on some images from the test set. Typically, true detections and false positives produce different types of traces. For instance, the trace signatures of false positives tend to have multiple complete chains and generate larger subtrees S. The traces generated by actual faces are usually more locally concentrated with fewer long chains. These phenomena are manifested in the learned trace models, which is why the likelihood ratio test is efficient in reducing false positives while maintaining faces.

Some comparisons with the baseline CTF system as well as other face detection methods are reported in Table 1. A detection rate of 88.8% with 126 false positives is achieved on 164 images from the test set. For the same detection rate, the false positive rate for the trace-based system is lower than that of the baseline CTF system. The trace-based results are also comparable to other well-known systems. It should be noted that the results from each system are reported on slightly different subsets of the CMU+MIT test set. Also, the performance of both the baseline CTF system and the trace-based system could very likely be improved by considering a richer feature set and/or a richer training set.

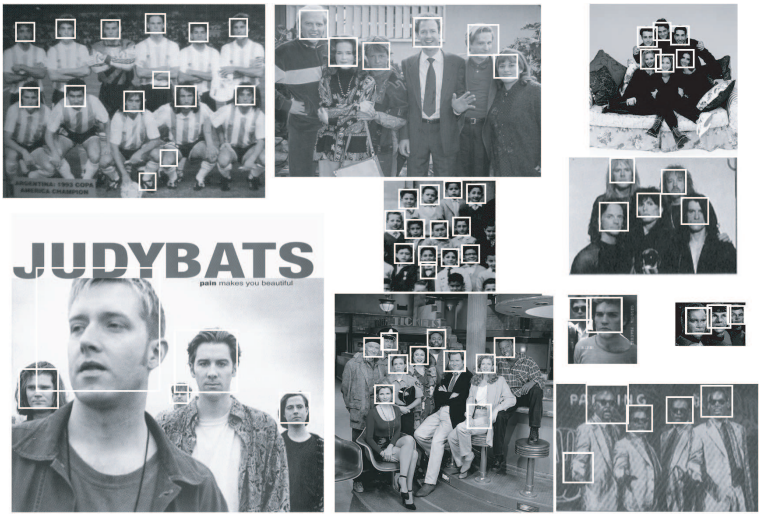

Fig. 5. Detection results on the CMU+MIT test set.

Table 1. Detection rates for various face detection systems

	Detection	False positives / image
Trace-Based CTF	89.1%	0.77
Baseline CTF	89.1%	1.11
Viola-Jones	90.8%	0.73
Rowley-Baluja-Kanade	89.2%	0.73

5.4 Comparison of ROC Curves

Another way to compare the baseline and trace-based systems is using ROC curves. The ROC principle is based on varying a free parameter (e.g., a threshold) in order to quantify the tradeoff between false positive and false negative error rates. In the case of the baseline CTF system, the parameter τ is the minimum number of complete chains required for an 8×8 window to be classified as containing the location of a face. For example, when $\tau = 1$ the detector produces a high number of false positives but few missed detections. For the trace-based system, the parameter is the threshold on the likelihood ratio, as discussed earlier. If multiple chains yield likelihood ratios above the threshold, the trace-based system chooses the chain (for that 8×8 block) that maximizes the likelihood ratio. For selected values of these thresholds, the false alarm rate and the true detection rate on the CMU+MIT test set are plotted in Fig. 7. The trace ROC curve is strictly superior, indicating that information does indeed reside in the "history of processing."

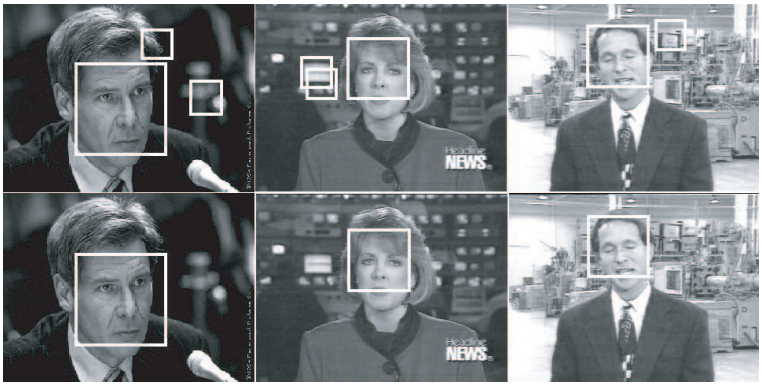

Fig. 6. Top row: The results of pure detection using the baseline CTF system. Bottom row: False positives are eliminated by setting an appropriate threshold on the trace likelihood.

6 Application to Face Tracking

Face tracking usually involves characterizing the temporal evolution of shapes, features or statistics. Tracking might be keyed by low level features such as color [16] and contours [17]. In some model-based methods [18], foreground regions are segmented by constantly updating a background model. Monte Carlo methods [19] applied to the posterior probability distribution of the object state employ dynamic sampling and Bayesian inference to estimate parameters of interest. Non-parametric methods, such as the mean-shift algorithm [20], have also been proposed for visual tracking. Most of these approaches exploit the temporal correlation between successive frames in order to refine the localization of a target. In most cases, real-time performance is achieved by restricting the search space by way of a highly constrained motion model. In general, work in face tracking has progressed largely independently from work in face detection and only a few approaches have attempted to merge them into a single framework [21].

In order to make inferences about a dynamical system, it is customary to specify two models – one that describes the evolution of the state with time (the system model) and one that relates the noisy measurement to the state (the measurement model). In our work, the state at time t is the set of poses of the faces in frame t and the trace is the measurement. A simple joint Markov model provides a natural probabilistic formulation and allows for the updating of information based on new measurements.

6.1 A Model for Face Tracking

In order to illustrate the role of the trace model, we shall only discuss tracking a single face, assumed visible throughout the sequence. We use $\mathbf{I}_{0:t-1}$ and $\theta_{0:t-1}$

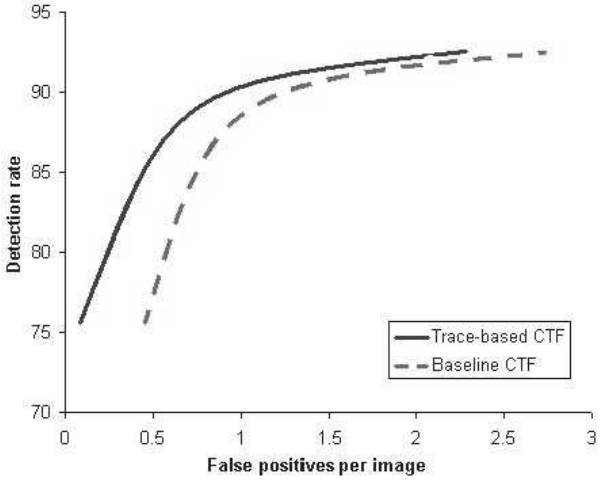

Fig. 7. ROC curve - detection rate vs. false positives on the MIT+CMU test set for the baseline CTF and the trace-based system

to denote the set of observed image frames and the set of observed poses, respectively, from time 0 to $t-1$. The (global) trace for image frame $\mathbf{I}_t$ is denoted by Z_t; recall from Section 4.2 that $Z_t = \{Z_t(i)\}$, where $Z_t(i)$ is the trace for the hierarchy corresponding to the i'th block. The tracking problem is formulated by estimating the pose of a face for every time t, given (i) a new trace, Z_t; (ii) the previously recorded set of traces, $Z_{0:t-1}$; and (iii) the set of previously observed poses, $\theta_{0:t-1}$. The MAP estimate $\hat{\theta}_t$ of the pose at time t is

$$\begin{aligned}\hat{\theta}_t &= \arg\max_{\theta_t} P(\theta_t | Z_{0:t}, \theta_{0:t-1}) \\ &= \arg\max_{\theta_t} \frac{P(Z_{0:t}, \theta_{0:t})}{P(Z_{0:t}, \theta_{0:t-1})} \\ &= \arg\max_{\theta_t} P(Z_{0:t}, \theta_{0:t}) \\ &= \arg\max_{\theta_t} P(Z_t, \theta_t | Z_{0:t-1}, \theta_{0:t-1})\end{aligned}$$

where we have rearranged the terms and dropped those independent of the argument θ_t. The trace and the pose are assumed to be a joint Markov process $(Z_t, \theta_t), t \geq 0$. The maximization is then simplified to

$$\begin{aligned}\hat{\theta}_t &= \arg\max_{\theta_t} P(Z_t, \theta_t | Z_{t-1}, \theta_{t-1}) \\ &= \arg\max_{\theta_t} P(Z_t | Z_{t-1}, \theta_t, \theta_{t-1}) P(\theta_t | Z_{t-1}, \theta_{t-1}).\end{aligned}$$

We further assume that the trace Z_t is conditionally independent of the previous trace and the previous pose given the current pose θ_t, and that the current pose

Fig. 8. Top row: The result of our tracker in three different frames. Bottom row: The raw results of pure detection in the same three frames.

θ_t is independent of the previous trace Z_{t-1} given the previous pose θ_{t-1}. These assumptions are reasonable and are consistent with other probabilistic-based tracking approaches. This leads to the following baseline tracker

$$\hat{\theta}_t = \arg\max_{\theta_t} P(Z_t|\theta_t)P(\theta_t|\theta_{t-1}). \tag{7}$$

The likelihood function $P(Z_t|\theta_t)$ of the global trace $Z_t = \{Z_t(i), i = 1, ..., n\}$ is defined in the same way as in Section 5.2, but under the simplifying constraint that all but one of the components of Θ represent "background". Writing $W(i(t))$ for the (unique) block containing the location component of θ_t, this likelihood can be written:

$$P(Z_t|\theta_t) = C(Z_t) \times \frac{P(Z_t(i(t))|\theta_t)}{P(Z_t(i(t))|B)} \tag{8}$$

where

$$C(Z_t) = \prod_{i=1}^{n} P(Z_t(i)|B)$$

is independent of θ_t.

A new track is initialized by examining the likelihood ratio as before, i.e., the maximization can be restricted to those θ_t which fall inside terminal pose cells at the end of complete chains. An old track is continued by restricting the pose space to regions in a neighborhood of the previous pose θ_{t-1}. The size of the neighborhood is determined by the variability captured by the pose transition model. The restriction of the pose-space does not limit the ability of the tracker to handle faces with large motions; these faces are detected as new faces by the CTF detection scheme.

The transition probability $P(\theta_t|\theta_{t-1})$ is assumed stationary and captures our prior knowledge about how the pose moves from one frame to another. Our

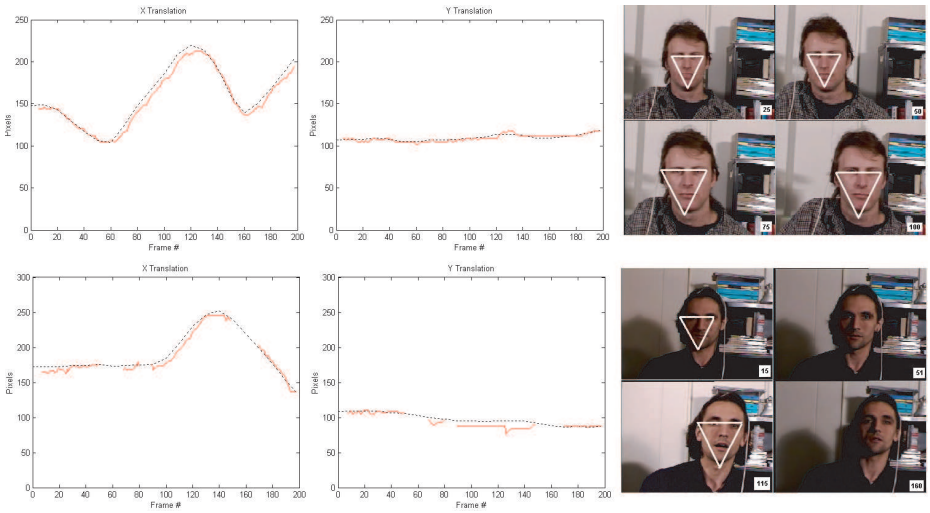

Fig. 9. Trajectories for the x and y coordinates of the estimated position during a tracking sequence. The dashed line represents ground truth and the solid line is the outcome of the trace-based Markov tracker. The right panel illustrates the results in some static frames extracted from the sequence.

transition model is learned from a set of training video sequences, recorded in a video conference setting with a subject normally seated not far from a fixed camera; there is then limited motion of the subject's face. The training sequences are manually landmarked and provide ground truth data for estimating pose transitions. A histogram of the pose differences $\theta_t - \theta_{t-1}$ is generated for the entire training set and serves as a good estimate for the pose transition model $P(\theta_t|\theta_{t-1})$.

Multiple faces and varying numbers of faces can also be accommodated since the evaluation of the trace is global. Multiple faces are tracked by implementing the baseline tracker independently for each new face. We omit the details concerning the initialization of new tracks and the removal of existing ones. Extending the algorithm to accommodate more variations in the pose of a face is straightforward. Pose hierarchies corresponding to left and right profiles are learned separately and are added directly to the original hierarchy (frontal faces) via a virtual node at the root and pose representation is augmented by a parameter indicating whether the view is frontal, left profile or right profile.

6.2 Results

Video sequences from commercial films and the Web are used to test the performance of the tracker. The sequences contain multiple faces per frame under various conditions of illumination and occlusion. With a standard desktop PC and with no MMX optimizations, faces are tracked at around 15 frames per

Fig. 10. Tracking of multiple faces: occlusion handling.

second. Since the evaluation of trace likelihoods is restricted to regions of interest, the speed of the tracker is mainly determined by the efficiency of detection. Real-time performance can be obtained by only executing the full-image detector every few frames or by incorporating global temporal information.

Fig. 6 illustrates the difference in the quality of single-frame detection between the trace-based Markov tracking model and the static algorithm (without the trace model) in [11]. Naturally, exploiting temporal continuity and the trace model removes false detections. In fact, tracking generally results in both a higher detection rate and a lower false positive rate. A higher detection rate is achieved because of the tracker's ability to "interpolate" when the detector fails to signal an alarm. The interpolation is possible due to the trace model's ability to produce valid probability measures even for poses that do not correspond to detected alarms. This phenomenon is mainly observed in cases where a subject temporarily violates the pose requirements or in cases of temporary occlusion. The state estimation of the Markov model filters out false positives which normally appear as high-frequency noise throughout a video sequence.

An empirical analysis of the tracker's performance is illustrated in Fig. 9. A single face is tracked in each sequence and its image coordinates are plotted through a segment of 200 frames. The video sequences are provided by [22] and are available at *http://www.cs.bu.edu/groups/ivc/HeadTracking/*. The frames in the right panel of Fig. 9 illustrate the result of the tracker at different points throughout the sequence. The dashed line represents ground truth which is obtained by manually landmarking the video sequence. The solid line is the outcome of the trace-based Markov tracker. As can be observed, the face position is correctly determined through most of the sequences. Some discontinuities are observed and are attributed to a failure of the CTF detection algorithm. The second sequence in Fig. 9 exhibits varying illumination; as a result, the detector provides inconsistent initialization and this propagates to the tracker, generating the observed discontinuity. A slight amount of jitter in position is attributed to inability of the first-order Markov model to integrate information over multiple frames.

Fig. 10 shows the result of tracking multiple faces through occlusions. Fig. 11 depicts the result of tracking a subject in a very challenging video sequence [23]. The face of the subject is successfully tracked despite heavy camera panning and unsteady focus. Unlike most tracking algorithms, the search is global and

Fig. 11. Tracking results on a difficult sequence with high camera instability.

the influence of the CTF detection model reduces the dependence on accurate motion estimation.

7 Conclusions

We have characterized the online computational process of an object detection system in the context of a graphical model for the history or "trace" of processing. This introduces a generative component into sequential detection strategies based on coarse-to-fine processing of a hierarchy of classifiers. The trace model captures and exploits the interactions among various classifiers within the hierarchy.

The utility of the trace model is demonstrated with experiments in face detection and tracking. There is a substantial gain in selectivity. Roughly speaking, at the same detection rate, the trace model eliminates around 40% of the false positives in deterministic hierarchical search. It also provides a unified framework for static face detection and dynamic face tracking, in which frame-based trace measures are merged with time-varying pose parameters within a simple Markov model. Unlike traditional tracking algorithms, there are no restrictions on the motion of a face. This is possible due to the computational efficiency of CTF detection, allowing for a nearly real-time search for multiple faces over an entire video frame at each instant. Further experiments will appear in forthcoming work.

References

1. R. Duda, P. Hart and D. Stork , " Pattern Classification," *John Wiley and Sons*, 2001.
2. M. Burl and P. Perona, "Recognition of planar object classes," *IEEE Proc. CVPR*, pp 223-230, 1996.
3. Y. Amit, "2D Object Detection and Recognition," *MIT Press*, 2002.
4. S. Krempp, D. Geman and Y. Amit, "Sequential learning with reusable parts for object detection," Technical Report, Johns Hopkins University, 2002.
5. S. Geman, D. Potter and Z. Chi, "Composition systems," *Quaterly of Applied Mathematics*, LX:707-736, 2002.

6. V. Vapnik, "The Nature of Statistical Learning Theory," *Springer-Verlag*, 1995.
7. E. Osuna, R. Freund and F. Girosi, "Training support vector machines: an application to face detection," *Proc. IEEE CVPR*, pp. 130-136, 1997.
8. H. Rowley, S. Baluja and T. Kanade, "Neural network-based face detection," *IEEE Trans. PAMI*, Vol. 20, pp. 23-38, 1998.
9. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Proc. CVPR*, 2001.
10. G. Blanchard and D. Geman, "Sequential testing designs for pattern recognition," *Annals of Statistics*, Vol. 33, pp. 155-1202, June 2005.
11. F. Fleuret and D. Geman, "Coarse-to-fine face detection," *IJCV*, Vol. 41, pp. 85-107, 2001.
12. Y. Amit, D. Geman and X. Fan, "A coarse-to-fine stratergy for pattern recognition," *IEEE Trans PAMI*, Vol. 26, no. 12, pp. 1606-1621, 2004.
13. H. Sahbi, "Coarse-to-fine support vector machines for hierarchical face detection," *PhD thesis*, Versailles University, 2003.
14. J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", *Morgan Kaufmann*, 1988.
15. K. Sung and T. Poggio, "Example-based learning for view-based face detection," *IEEE Trans. PAMI*, Vol. 20,pp. 39-51, 1998.
16. K. Schwerdt and J. Crowley, "Robust face tracking using colour", *Proc. Int'l Conf. Auto. Face and Gesture Recognition*, pp. 90-95. 2000.
17. D. Decarlo and D. Metaxas, "Deformable model based face shape and motion estimation," *Proc. Int'l Conf. Auto. Face and Gesture Recognition*, 1996.
18. C.J. Edwards, C.J. Taylor and T.F Cootes, "Learning to identify and track faces in an image sequence," *Proc. Int'l Conf. Auto. Face and Gesture Recognition*, pp 260-265, 1998.
19. M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *IJCV*, vol. 29, pp 5-28, 1998.
20. D. Comaniciu, V. Ramesh and P. Meer, "Kernel based object tracking," *IEEE Trans. PAMI*, Vol. 25. pp:564-577, 2003.
21. B. Li and R. Chellappa, "A generic approach to simultaneous tracking and verification in video," *IEEE Trans. Image Processing*, 11:530-544, 2002.
22. M.L. Cascia, S. Sclaroff and V. Athitos, "Fast reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *IEEE. Trans. PAMI*, Vol. 21, No. 6, 1999.
23. http://www.madonnalicious.com/downloads.html