Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Vision and Image Understanding 98 (2005) 182–210
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

A generative framework for real time object detection and classification☆

Ian Fasel[a,b,*], Bret Fortenberry[a], Javier Movellan[a]

[a] *Institute for Neural Computation, University of California, San Diego, USA*
[b] *Department of Cognitive Science, University of California, San Diego, USA*

Received 27 July 2004; accepted 27 July 2004

Abstract

We formulate a probabilistic model of image generation and derive optimal inference algorithms for finding objects and object features within this framework. The approach models images as a collage of patches of arbitrary size, some of which contain the object of interest and some of which are background. The approach requires development of likelihood-ratio models for object versus background generated patches. These models are learned using boosting methods. One advantage of the generative approach proposed here is that it makes explicit the conditions under which it is optimal. We applied the approach to the problem of finding faces and eyes on arbitrary images. Optimal inference under the proposed model works in real time and is robust to changes in lighting, illumination, and differences in facial structure, including facial expressions and eyeglasses. Furthermore, the system can simultaneously track the eyes and blinks of multiple individuals. Finally we reflect on how the development of perceptive systems like this may help advance our understanding of the human brain.

Keywords: Blink detection; Eye detection; Boosting; Generative models

☆ This research was funded by NSF IIS-0220141, NSF IIS-0329287, NSF IIS-0086107, and UC DIMI-01-1030.
* Corresponding author.
E-mail addresses: ianfasel@mplab.ucsd.edu, ian@mplab.ucsd.edu (I. Fasel), bret@fortenberry.com (B. Fortenberry), movellan@mplab.ucsd.edu (J. Movellan).

1077-3142/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2004.07.014

1. Introduction

Since its official beginnings at the Dartmouth conference in 1956, the cognitive approach has become the dominant paradigm in the social sciences. Historically, the approach has fashioned many heated debates: early attention vs. late attention, working memory vs. short term memory, serial vs. parallel processing, analogical vs. propositional representations, symbolic vs. sub-symbolic processing, modular vs. interactive architectures. Unfortunately, many of these debates have turned out to be undecidable, contributed little to our understanding of human nature, and have had little impact on society at large (see Fig. 15).

Modern approaches and methods are needed that avoid scholastic debates. One approach which we have found particularly useful was originally proposed by Marr [29]. The approach focuses on understanding the nature of the problems faced by the brain and finding possible solutions to these problems [6]. When pursuing this endeavor we have found that probability theory, in particular the use of probabilistic generative models, was a fruitful analytical tool. The third author of this paper referred to this methodological stance as *probabilistic functionalism* [30]. One characteristic of probabilistic functionalism is the focus on solving specific problems under general conditions rather than solving abstract problems under restricted laboratory conditions. To focus simultaneously on the specificity of the problem and the generality of the solution is critical, otherwise one can easily get caught in frustrating theoretical debates or by trick solutions that inform us little about the brain. The current paper can be seen as an application of the methods of probabilistic functionalism to help understand the problem of eye and eye-blink detection. We do so by formulating an analytical model of the problem at hand, studying how optimal inference would proceed under such a model, and evaluating the performance of the optimal inference algorithm in natural conditions.

The study of face perception has been revitalized thanks to recent progress in cognitive neuroscience. The advent of modern neuro-imaging is revolutionizing the study of the mind and presenting a picture of the human brain far different from a general purpose computing machine. Single neuron recording and imaging studies are showing specific neural systems that play a crucial role in the perception of faces, facial features, and facial expressions. These include the fusiform face area, superior temporal sulcus, orbital frontal cortex, frontal operculum, right somatosensory cortex, and the amygdala [25,16].

Face perception has been a traditional area of research in developmental psychology, a discipline that studies how the human mind develops from infancy to adulthood. Face processing in general and eye detection in particular is deemed so important in this field that some of its most influential researchers have postulated the need for innate eye detection and gaze processing modules. These ideas are still controversial but recent experiments have shown that from birth human infants are exceptionally sensitive to the eye and to mutual gaze engagement [10,23]. These systems may help tune the newborn infant towards interaction with their caregivers [1].

In recent years there has been an emerging community of machine perception scientists focused on automatic detection of faces and facial behavior. The special

importance of the eyes is becoming quite clear within this community. There are at least two reasons for this: (1) Proper registration. In a recent evaluation of state of the art face recognition system it was proposed that a large proportion of the failures of these system was due to poor alignment and registration of facial features, particularly in outdoors conditions. Good eye detection in realistic environments may thus have a tremendous impact on the accuracy of face perception technologies [31]. (2) Information value. Eyes and eye movements are a particularly important source of information in human interaction. Indeed, the Facial Action Coding System of Ekman and Friesen [8], arguably the most comprehensive standard for coding facial behavior, devotes 15 categories to describe eye behavior (see Table 1). Only the mouth surpasses the eyes in the number of categories assigned to it. This reflects the fact that eye behavior is extremely rich and particularly informative about the state of human beings.

Current work on eye detection divides into approaches based on visible spectrum cameras and approaches based on near-infra-red (NIR) cameras. In indoor and relatively controlled conditions the spectral properties of the pupil under NIR illumination provide a very clean signal that can be processed very fast and accurately [17,21,22]. While NIR based methods are practical and worth pursuing, it is also important to pursue visual spectrum methods for the following reasons: (1) NIR based methods tend to produce a large number of false positives when used in relatively uncontrolled illumination conditions; (2) NIR based methods do little to further our understanding about the perceptual problem the brain solves when processing faces in natural conditions.

Of all the eye related behaviors perhaps the most important is blinks, action unit 45 in the Facial Actions Coding System. This is due to its relevance in several fields,

Table 1
FACS codes involving eyes

Code	Descriptor	Muscles involved	Example
AU5	Upper lid raiser	Levator palpebrae superioris	
AU6	Cheek raiser	Orbicularis oculi, pars orbitalis	
AU7	Lid tightener	Orbicularis oculi, pars palebralis	
AU41	Lid droop	Relaxation of levator palpebrae superioris	
AU42	Slit	Orbicularis oculi	
AU43	Eyes closed	Relaxation of levator palpebrae superioris; orbicularis oculi, pars palpebralis	
AU44	Squint	Orbicularis oculi, pars palpebralis	
AU45	Blink	Relaxation of levator palpebrae superioris; orbicularis oculi, pars palpebralis	
AU46	Wink	Relaxation of levator palpebrae superioris; orbicularis oculi, pars palpebralis	
AU61	Eyes turn left	Lateral and medial rectus	
AU62	Eyes turn right	Lateral and medial rectus	
AU63	Eyes up	Superior rectus	
AU64	Eyes down	Inferious rectus	
AU65	Walleye	Lateral rectus	
AU66	Crosseye	Medial rectus	

including neurology, physiology, and psychology. For example, blink rate is known to vary with physiological and emotional arousal, cognitive effort, anxiety, fatigue, and deceit [18,7,24,36,21]. Ji and Yang [22] presents a state of the art method to detect blinks in real time using NIR imaging. Approaches based on visual spectrum images also exist. Bartlett et al. [2] present an approach to detect blinks in indoors environment using Support Vector Machines. Cohn et al. [4] describe an approach that uses hand-coded eye-blink detectors. They report results comparable to those of Bartlett et al. [2] on the same testing dataset. Both systems handled out-of-plane rotations of the head by fitting a 3D deformable model of the head and then re-rendering the image into a frontal view.

2. A generative model for images

In this section, we frame the problem of finding faces and facial features as a Bayesian inference problem: We formulate a model of how images are generated and then derive an algorithm for making optimal inferences under this model. One advantage of generative models is that probability estimates of the categories of interest are computed explicitly, facilitating integration with other potential sources of information not necessarily considered at design time. In addition generative models force us to make our assumptions explicit, facilitating progress towards more effective algorithms.

Unless otherwise stated, capital letters will represent random variables and small letters specific values taken by those variables. When possible we use informal shorthand notation and identify probability functions by their arguments. For example, $p(y)$ is shorthand for the probability (or probability density) that the random matrix Y takes the specific value y.

We model the image as a collage of rectangular patches of arbitrary size and location, some patches rendering the object of interest, the others rendering the background. Given an image our goal is to discover the patches that rendered the object. Let Y be a random matrix representing an image with a fixed number of pixels. Let y be a specific sample from Y. Let $\mathscr{A} = (a_1, a_2, \ldots, a_n)$ be an enumeration of all possible rectangular image patches, e.g., a_i determines the position and geometry of a rectangle on the image plane. Let y_{a_i} be a matrix whose elements are the values of y for the pixels in the rectangle a_i. Let $H = (H_1, \ldots, H_n)$ be a random vector that assigns each of the n patches to one of three categories: H_i takes the value 1 when the patch a_i renders the object of interest, it takes value -1 when it renders the background, and value 0 when it is not rendered (see Figs. 1 and 2).

The image generation process proceeds as follows (see Fig. 1). First a segmentation h is chosen with probability $p(h)$. Then for each patch a_i if $H_i = 1$ then an image of the size of a_i is chosen from the object distribution $q(\cdot|a_i,1)$ independently of all the other patches. If $H_i = -1$ then a background image y_{a_i} is chosen from the background distribution $q(\cdot|a_i,-1)$. If $H_i = 0$ then a_i is not rendered. The observed image y is the collection of the rendered patches.

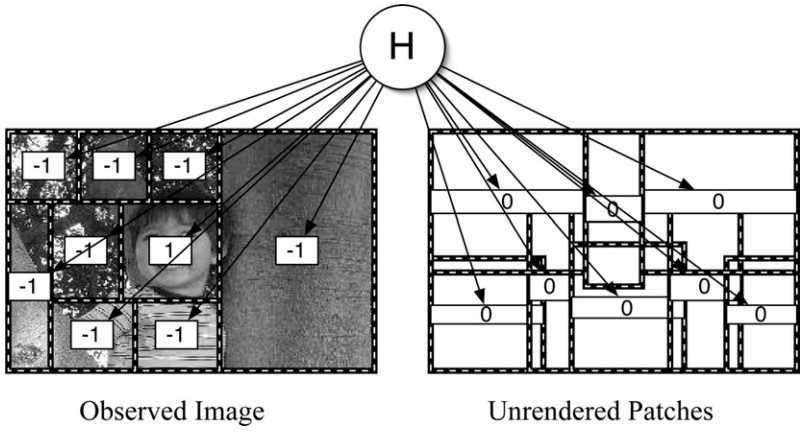

Fig. 1. The hidden variable H determines which image patches will render the background (-1) which patches will render the object of interest (1) and which patches will not be rendered (0). The set of rendered patches determine the observed image.

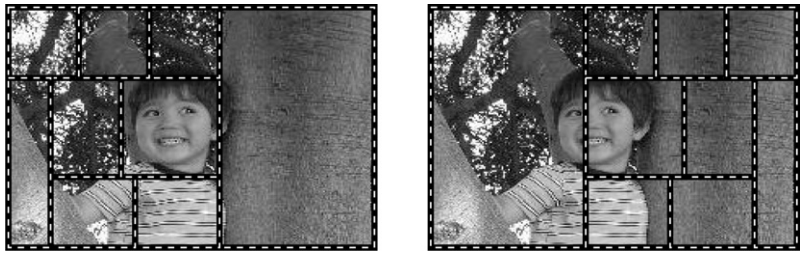

Fig. 2. The segmentation on the left contains the patch that generated the object of interest (i.e., the face). It will be hard for this segmentation to explain the image as a collection of background patches. The segmentation on the right does not contain the object patch. Since the background model includes wrongly shifted versions of faces it will be easy to explain the image as a collection of background patches.

The model is specified by the prior probabilities $p(h)$ and by the object and background rendering distributions q. The prior is specified by the marginal probabilities $\{P(H_i = 1) : i = 1, \ldots, n\}$, with the constraint that values of h that do not partition the image plane have zero probability, and by one of the two following constraints: (I) For cases in which we know there is one and only one object of interest on the image plane, only values of h with a single 1 are allowed. (II) For cases in which there may be an arbitrary number of objects of interest we assume the location of a rendered object does not inform us about the location of other objects, expect for the fact that each pixel can only be rendered by a single object or background element. More formally, for $i = 1, \ldots, n$, the random variables $\{H_j : j \neq i\}$ are independent of H_i when conditioning on the event $\{H_i \neq 0\}$. For a given image y our goal is to detect patches rendered by the object. There are two cases of interest: (I) We know there is one and only one patch rendered by the object. (II) There is an unknown and arbitrary number of patches rendered by the object model.

2.1. Case I: single object

Suppose we know there is one and only one patch in the image plane that rendered the object of interest. Then our goal is to find the most probable patch $\hat{k} \in \{1, \ldots, n\}$ given the image y, i.e.,

$$\hat{k} = \operatorname*{argmax}_i P(H_i = 1|y). \tag{1}$$

Using the law of total probability we have that

$$P(H_i = 1|y) = \frac{\sum_h P(H_i = 1)p(h|H_i = 1)p(y|h, H_i = 1)}{p(y)}. \tag{2}$$

Note that $p(h|H_i = 1)$ is zero if the segmentation h contains the patch a_i and one otherwise. Moreover, for any h that includes a_i we have that

$$p(y|h, H_i = 1) = \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} Z(h, y), \tag{3}$$

where

$$Z(h, y) = \prod_{i:h_i \neq 0} q(y_{a_i}; a_i, -1). \tag{4}$$

The term $Z(h, y)$ describes how well the image y can be explained by the segmentation h with all the patches rendering background, no objects. Thus

$$\begin{aligned} P(H_i = 1|y) &= P(H_i = 1) \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} \frac{\sum_h p(h|H_i = 1)Z(h, y)}{p(y)} \\ &= P(H_i = 1) \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} \frac{E(Z(H, y)|H_i = 1)}{p(y)}. \end{aligned} \tag{5}$$

The term $E(Z(H, y)|H_i = 1)$ represents how well the image y can be explained as a mosaic of background patches, provided one of those patches is a_i. If the background distribution model $q(\cdot|a_k, -1)$ includes wrongly shifted and scaled versions of the object of interest then $E(Z(H, y)|H_i = 1)$ should be small for the patch that actually rendered the object, and large otherwise. This is due to the fact that the patch that includes the object will be hard to explain by the background model (see Fig. 2). More formally if $E(Z(H, y)|H_{\hat{k}} = 1) \leqslant E(Z(H, y)|H_i = 1)$ for $i = 1, \ldots, n$ then

$$\begin{aligned} \hat{k} &= \operatorname*{argmax}_i P(H_i = 1|y) = \operatorname*{argmax}_i P(H_i = 1) \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} \\ &= \operatorname*{argmax}_i \log P(H_i = 1) + \log \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)}. \end{aligned} \tag{6}$$

The optimal inference algorithm prescribes scoring all possible patches in terms of a function that includes the prior probability of that patch containing an object and a likelihood ratio term. The patch that maximizes this score is then chosen.

2.2. Case II: multiple objects

This case applies, for example, in face detection problems for which we do not know a priori how many faces may appear on the image plane. To formalize the problem we define a function Φ measuring the degree of match between any two arbitrary segmentations h and h':

$$\Phi(h,h') = \sum_{i=1}^{n} \rho(H_i, H'_i), \tag{7}$$

$$\rho(H_i, H'_i) = (\delta_{H_i,1} + \delta_{H_i,-1})\delta_{H_i,H'_i}, \tag{8}$$

where δ is the Kroenecker delta function. ρ counts the number of patches for which both h and h' assign the same "object" or "background" label and ignores all the patches that are not rendered by h. Our goal is to find a partition $\hat{h}$ that optimizes the expected match

$$\hat{h} = \underset{h'}{\operatorname{argmax}} E(\Phi(H,h')|y) = \sum_{h} p(h|y)\Phi(h,h'). \tag{9}$$

The optimal assignment follows:

$$\hat{h}_i = \begin{cases} 1 & \text{if } p(H_i = 1|y) > p(H_i = -1|y), \\ -1 & \text{else.} \end{cases} \tag{10}$$

Thus, to find the optimal assignment we need to scan all possible image patches $a_1, \ldots, a_n$, compute the log-posterior probability ratio

$$\log \frac{P(H_i = 1|y)}{P(H_i = -1|y)}, \tag{11}$$

and assign "object" labels to the patches for which this ratio is larger than 0.

Using the law of total probability we have that

$$P(H_i = 1|y) = \sum_{h} P(H_i = 1)p(h|H_i = 1)p(y|h, H_i = 1), \tag{12}$$

where $p(h|H_i = 1)$ is zero if the segmentation h contains the patch a_i, one otherwise, and

$$p(y|h, H_i = 1) = q(y_{a_i}; a_i, 1) \prod_{j \neq i : h_j \neq 0} q(y_{a_j}; a_j, h_j). \tag{13}$$

Thus for $k = -1, 1$ we have that

$$P(H_i = k|y) = P(H_i = k)q(y_{a_i}; a_i, k) \sum_{h} p(h|H_i = k) \prod_{j \neq i : h_j \neq 0} q(y_{a_j}; a_j, h_j) \tag{14}$$

and due to the fact that $\{H_j; j \neq i\}$ are independent of H_i given $\{H_i \neq 0\}$ it follows that

$$\log \frac{P(H_i = 1|y)}{P(H_i = -1|y)} = \log \frac{P(H_i = 1)}{P(H_i = -1)} + \log \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)}. \tag{15}$$

To make optimal inferences all we need is a model for the prior probability of object locations and a model for the log-likelihood ratios of image patches of arbitrary geometry. In Section 3, we will see how these models can be learned using boosting methods.

3. Learning likelihood ratios using gentleboost

The inference algorithm presented above requires a likelihood ratio model. Given an arbitrary image patch y we need an estimate for the ratio between the probability of such a patch being generated by the object class vs the background class. In this paper, we learn these likelihood ratios using GentleBoost, a boosting algorithm developed by Friedman et al. [14]. Boosting [13,12] refers to a family of machine learning algorithms for learning classifiers by sequential accumulation of experts that focus on the mistakes made by previous experts. Friedman et al. [14] showed that boosting methods can be reinterpreted from the point of view of sequential statistical estimation, an interpretation that makes it possible to use it in the generative framework proposed here.

The goal is to learn a model for the log-likelihood ratio of arbitrary image patches. During training we are given a set of examples $\{(y_i, z_i): i = i, \ldots, m\}$, where y_i is an image patch, and $z_i \in \{-1, +1\}$ its category label, i.e., object or background. The model used in GentleBoost is of the following form:

$$p(y) = \frac{1}{1 + e\{-2\sum_j f_j(y)\}}, \tag{16}$$

where $p(y)$ is the probability that image patch y belongs to one of the two categories of interest, and $f_i(y)$ is the opinion of the ith expert, as defined in Fig. 3. GentleBoost can be seen as an application of the Newton–Raphson optimization algorithm to the problem of minimizing the following χ^2 error [14]

$$\rho = \sum_i \frac{t_i - p(y_i)}{\sqrt{p(y_i)(1 - p(y_i))}}, \tag{17}$$

where $t_i = 0.5(z_i + 1) \in \{0,1\}$ is the category label for the ith training input y_i. Since $p(y_i)$ is the probability of a Bernoulli random variable with mean $p(y_i)$ and standard deviation $\sqrt{p(y_i)(1 - p(y_i))}$, then ρ can be seen as the number of standard deviations between the observed label and the average label value. As the number of examples in the training set increases, minimizing the χ^2 error becomes identical to maximizing the likelihood. However, when the number of samples is small, χ^2 estimators can be more efficient than maximum likelihood estimators.

3.1. Selecting wavelets and tuning curves

GentleBoost chooses a set of experts $f_1, f_2, \ldots$ in a sequential manner. Each Newton–Raphson step results on the selection of the expert that maximally reduces the current χ^2 error given the already selected set of experts. In practice this can be done in a variety of ways. We use the following approach:

- Let $\{ (y_i, z_i) : i = i, \cdots, m\}$, be a set of training examples, where y_i is the an image patch, and $z_i \in \{-1, +1\}$ its category label.
- Let $P_t(i)$ represent the weight assigned to the i^{th} examples at the beginning of the t iteration of the GentleBoost algorithm.
- Let the initial distribution be as follows: $P_0(i) = 1/m$, for $i = 1, \cdots m$, i.e., each training example is weighted equally.
- For time $t = 1, \cdots$
  - For wavelet $w = 1, c \ldots, n$
    Use kernel-regression to find the tuning curve h that best fits the set of triplets $\{(w(y_i), z_i, P_t(i)) : i = 1, \cdots, m\}$.
  - Choose $(\hat{w}, \hat{f})$ the wavelet and tuning curve that minimize the errof function ρ. They define the expert selecte at iteration t

$$f_t(y) = \hat{h}(\hat{w}(y))$$

  - Update the distribution over training elements

$$P_{t+1}(i) = P_t(i)\frac{e^{-f_t(y_i)z_i}}{Z_t}$$

    where Z_t is a normalization factor

$$Z_t = \sum_i P_t(i) e^{-f_t(y_i)z_i}$$

  - Update the posterior probability model

$$p(y) = \frac{1}{1 + e^{-2\sum_{n=1}^{t} f_n(y)}}$$

Fig. 3. The GentleBoost approach used in this paper.

We start with a large pool of wavelets $\{w_1, \ldots, w_n\}$, about 170,000 in our case (see Section 5), and define an expert as the combination of a wavelet w and a tuning curve h to be defined below. By iteration t of the Newton–Raphson method, we have already selected $t-1$ experts. At this point we go over each wavelet w in our pool and for each wavelet we estimate the tuning function $h : \mathscr{R} \mapsto [-1, 1]$ that minimizes ρ given the outputs of the wavelet w and the information provided by the $t-1$ experts already selected. This function can be shown to have the following form:

$$h(w(y)) = E^{P_t}[Z|w(y)], \tag{18}$$

where $Z \in \{-1, 1\}$ is the category label, and the expectation is taken with respect to the distribution induced by the weights assigned by GentleBoost to the different

training data (see Fig. 3). We estimate the function h using the Nadaraya–Watson kernel regression method for density estimation [34]. The training examples used in this regression method are the set of triplets $\{(w(y_i), z_i, P_t(y_i)) : i = 1, \ldots, m\}$, where $w(y_i)$ is the regressor variable, z_i the label we wish to predict, and $P_t(y_i)$, the weight of example y_i, z_i).

We call the function h the *tuning curve* for the wavelet w. After we find the optimal tuning curves for all the wavelets in the original pool, we choose the wavelet $\hat{w}$ and corresponding tuning-curve $\hat{h}$ that minimize ρ. This pair defines the expert selected for iteration t, i.e.,

$$f_t(y) = \hat{h}(\hat{w}(y)). \tag{19}$$

The process is iterated, each time adding a new wavelet and tuning curve, until ρ no longer decreases. This procedure is illustrated in Figs. 5 and 3.

By the end of training process we have a model for the posterior probability of the object class given arbitrary image patches y

$$p(y) = \frac{1}{1 + e\{-2\sum f_t(w_t(y))\}}. \tag{20}$$

This posterior probability estimate reflects the particular proportion π of examples of each class used during training. The inference algorithm in (22) requires log-likelihood ratios, not log-posteriors. These can be easily derived from (20) using Bayes rule

$$\begin{aligned} \log \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} &= \log\left(\frac{1-\pi}{\pi}\right) + \log\left(\frac{p(H_k = 1|y_{a_i})}{p(H_k = -1|y_{a_i})}\right) \\ &= \log\left(\frac{1-\pi}{\pi}\right) + 2f(x). \end{aligned} \tag{21}$$

Combining (6) and (20) we get

$$\hat{k} = \max_i p(H_i = 1|y) = \max_i \log p(H_i = 1) + 2f(y_{a_i}). \tag{22}$$

4. Situation based inference

One common approach to eye detection is based on the operation of a set of independent feature detectors [19,11]. The output of these detectors (e.g., a detector for the left eye, a detector for the right eye, a detector for the tip of the nose, etc.) is integrated by looking for configurations that match the distribution of interfeature distances typical of the human face [38,27,26]. Unfortunately this method scales exponentially with the number of false alarms of each feature detector. Suppose our goal is to find the center of an eye with 1 pixel accuracy. This requires for background models to include examples of eyes shifted by 1 pixel from the center position. In practice, a detector efficient at distinguishing eyes slightly shifted from center is also likely to produce a large number of false positives when scanning gen-

eral backgrounds that do not include faces, creating an unsurmountable problem for methods that rely on feature detection.

The approach we propose here is based on the idea of a bank of situational or context dependent experts operating at different levels of specificity. For example, since the eyes occur in the context of faces, it may be easier to detect eyes using a very large context that include the entire face and then formulate feature detectors specifically designed to work well under such context. While we may think of these as face detector, we can also think of them as eye detectors that happen to have very large receptive fields. This form of eye detection works under very general context conditions, avoiding the proliferation of false alarms, but provides poor information about the precise location of the eyes. These eye detectors are complemented by context-specific eye detectors that provide very precise information about the position of the eyes.

More formally, let y represent an observed image, S represent a contextual situation (e.g., the location and scale of a face on the image plane), and O represent the location of the left eye of that face on the image. Using the law of total probability we have that

$$p(o|y) = \int p(s|y)p(o|sy)\,\mathrm{d}h. \tag{23}$$

Here $p(s|y)$ works as a situation detector. Its role is to find regions in the image plane that are likely to contain eyes due to the fact that they contain faces. The $p(o|sy)$ term is a situation specific eye detector. For example it may work when the location and scale of the face on the image plane is known. In this example $p(s|y)$ partitions the image pixels into those belonging to the face, y_f, and those belonging to the background, y_b. Once the position and scale of the face are known, the background provides no additional information about the position of the eye, i.e.,

$$p(o|y_f, y_b, s) = p(o|y_f s). \tag{24}$$

The situational approach proposed here can be iterated, where one first detects a general context, followed by detection of a context within a context, each time achieving higher levels of precision and specificity allowed by the fact that the context models become smaller and smaller on each iteration.

5. Real-time system architecture

In the next sections we describe and evaluate an algorithm that performs optimal inference under the assumptions of the generative model described above. The current system utilizes two types of eye detectors: The first type, which can be thought of as a face detector, starts with complete uncertainty about the possible location of eyes on the image plane. Its role is to narrow down the uncertainty about the location of the eyes while operating in a very wide variety of illumination and background conditions. The second type of detector operates on the output of the first detector. As such it can assume a restricted context and achieve high location accuracy. Once the most likely eye location is chosen, the image patch surrounding the

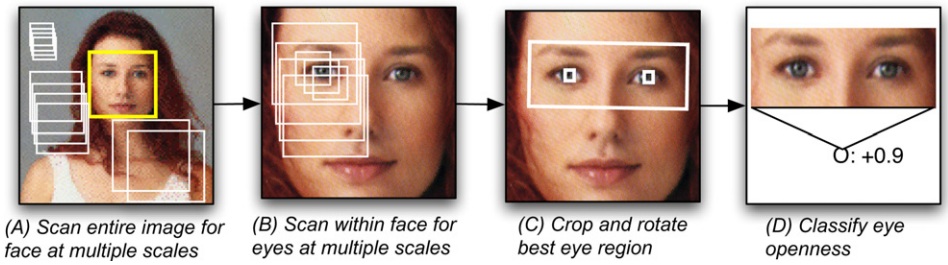

Fig. 4. Flowchart for face, eye, and blink detection.

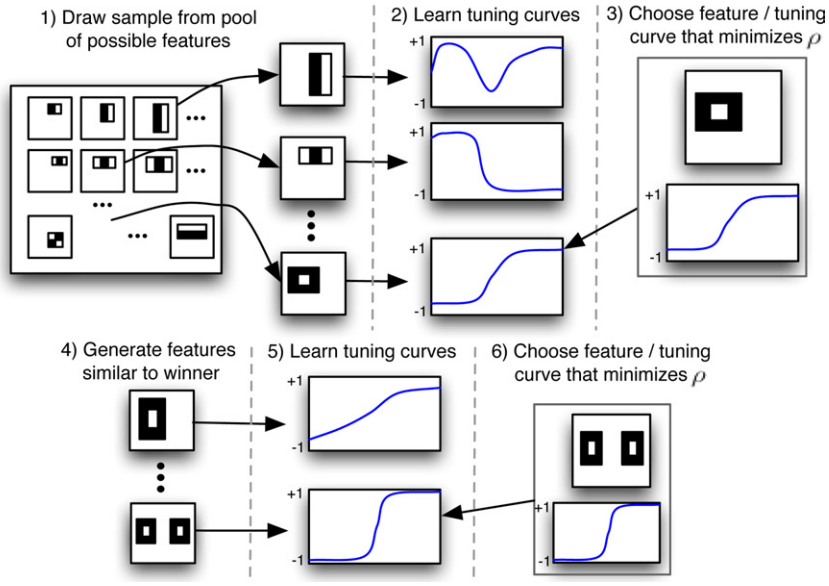

Fig. 5. Flowchart for one iteration of the feature selection procedure.

eyes is passed to a blink detection for analysis. The flowchart for this procedure is shown in Fig. 4.

While the system described here operates on video images in real time, it currently treats each frame as independent of the previous frames, making it equally useful for static images as for video. Treating each video frame independently allows the system to simultaneously code eye location and behavior on multiple faces that may come in and out of the scene at random times.

5.1. Stage I: eye detection in general background conditions

As described above the first component of the inference process locates regions of the image plane that contain faces, and thus eyes. This module operates under very

general background and illumination conditions and greatly narrows down the plausible locations of eyes on the image plane. It makes no prior assumptions about the location of the face.

The general procedure for the image search is similar to the multiscale search of Rowley et al. [32], who trained a single binary classifier to classify face vs. non-face for patches of fixed size (20×20 pixels), then used that classifier to classify all possible patches in the image. Faces larger than the original size were found by repeating the search in copies of the image scaled to smaller sizes (thus, a 20×20 pixel face in a 1/4 size copy of the image corresponds to an 80×80 pixel face in the corresponding location in the original).

We use a very similar scheme, however rather than a binary classifier, we developed a likelihood-ratio model using a dataset of Web images provided by Compaq Research Laboratories. This dataset contains 5000 images containing frontal upright faces taken under a variety of illumination conditions, facial expressions, facial hair, eyeglasses, hats, etc., of widely varying image quality. Faces were cropped and scaled to 24×24 pixels square. The negative examples were sampled from a dataset of 8000 images collected from the Web and known not to contain faces. Similarly, these images contained a wide variety of natural indoor and outdoor scenes, text, illustrations, posed images of objects, etc., with varying image quality. The advantage of this Web dataset is that it includes far more variability than most other closed databases.

Due to the multi-scale search, about 1 billion total patches are possible in these 8000 images. For the initial negative examples for training, 10,000 square patches, of arbitrary size and at arbitrary locations in the images, were sampled from this dataset. Patches were then scaled down to 24×24 pixels. The set of negative samples changes during training thanks to the bootstrap round (described below), so ultimately all 1 billion possible patches were used at some time during training (see Fig. 8).

The likelihood-ratio model was trained using the GentleBoost method described in Section 3. GentleBoost sequentially chooses wavelets from a large pool and combines them to minimize a χ^2 error function. The pool of wavelets we choose from was based on [37] and consists of Haar-like wavelets. The main reason for their use is that their output can be computed very fast by taking the sum of pixels in two, three, or four equal-sized, adjacent rectangles and taking differences of these sums. To this original set we add a center-surround type wavelets and mirror image wavelets that are sensitive to patches symmetric about vertical axis (see Fig. 7).

The GentleBoost approach described in Section 3.1 requires computing tuning curves on each of the wavelet candidates. It is very computationally expensive to perform an exhaustive search over all these wavelets—in a 24×24 pixel window, there are over 170,000 possible wavelets of this type. To speed up training, we break the wavelet selection step into two stages (see Fig. 5). First, at each round of boosting, we take a random sample of 5% of the possible wavelets. For each wavelet we find the tuning curve that minimizes the loss function ρ if that particular wavelet were added to the pool of already chosen wavelets. In step two, we refine the selection by finding the best performing single-wavelet classifier from a new set of wavelets

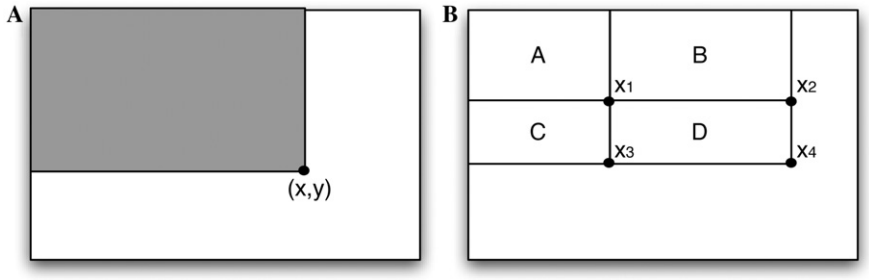

Fig. 6. The Integral Image: (A) The value of the pixel at (x, y) is the sum of all the pixels above and to the left. (B) The sum of the pixels within rectangle D in the original image can be computed from points in the integral image by $x_4 - x_2 - x_3 + x_1$.

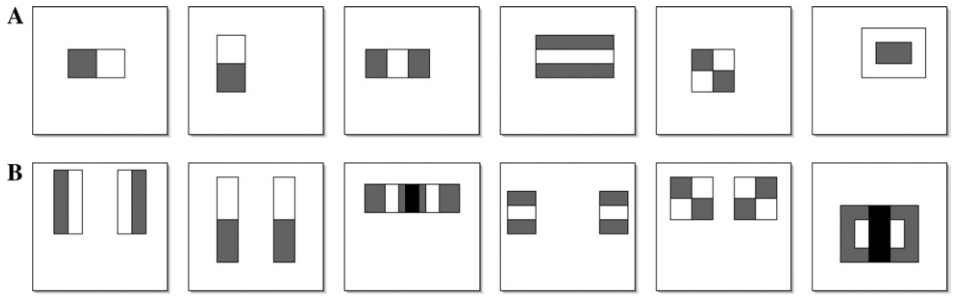

Fig. 7. Each wavelet is computed by taking the difference of the sums of the pixels in the white boxes and grey boxes. (A) Wavelet types include those in [37], plus a center-surround type wavelet. (B) In the refinement step, the same wavelet types superimposed on their reflection about the y axis are also possible.

Fig. 8. Examples of faces and non-faces used in training the face detector.

generated by shifting and scaling the best wavelet by two pixels in each direction, as well as composite wavelets made by reflecting each shifted and scaled wavelet horizontally about the center and superimposing it on the original. Using the chosen classifier as the weak learner for this round of boosting, the weights over the examples are then adjusted using to the GentleBoost rule. This wavelet selection process is then repeated with the new weights, and the boosting procedure continues until the performance of the system on a validation set no longer decreases.

The inference algorithm calls for likelihood ratio models at multiple scales. Likelihood ratios for larger image patches are obtained by linearly scaling the patches down to 24×24 pixels and then applying the likelihood ratio model trained on that particular scale. Thanks to the choice of Haar-like wavelets for the higher level image representation, this interpolation step can accomplished in constant time regardless of scale (see [37,33] for a more detailed explanation).

Following [37], rather than training a "monolithic" classifier which evaluates all its wavelets before it makes a decision, we divided the classifier into a sequence of smaller classifiers which can make an early decision to abort further processing on a patch if its likelihood-ratio falls below a minimum threshold. We can think of this as a situational cascade where each level of the cascade is trained only on patches

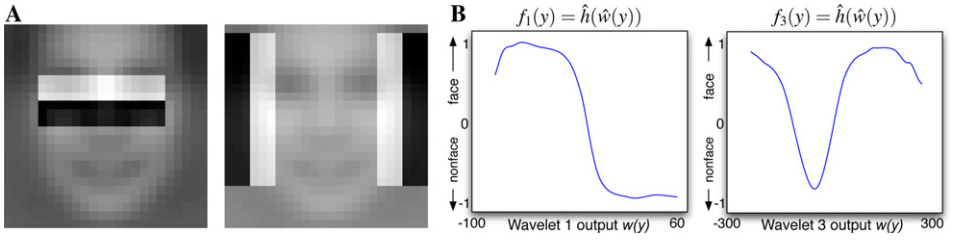

Fig. 9. The first two wavelets (A) and their respective tuning curves (B) for face detection. Each wavelet is shown over the average face. The tuning curves show the evidence for face (high) vs. non-face (low), as a function of the output of the wavelet, shown increasing from left to right. The first tuning curve shows that a dark horizontal region over a bright horizontal region in the center of the window is evidence for an eye, and for non-eye otherwise. The second tuning curve is bimodal, with high contrast at the sides of the window evidence for a face, and low contrast evidence for non-face.

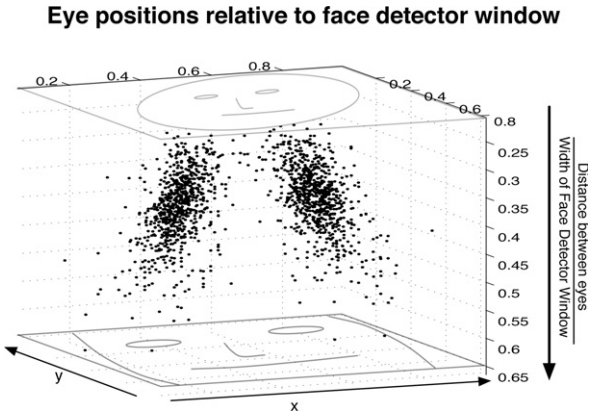

Fig. 10. The face detection window can vary from closely cropping the face (negative z-axis) to loosely cropping the face (positive z-axis). The points show typical eye locations relative to the face detection window over a sample database of face images. We model this variability with a three-dimensional Gaussian, where the x- and y-axes are space, and the z-axis is scale, i.e., ratio of distance between eyes to size of the face-detector window. We use this to model the prior probability of a location containing an eye given the face detection window.

that survived the previous levels. After each element of the cascaded is trained, a boot-strap round (*ala* Sung and Poggio [35]) is performed, in which the full system up to that point is scanned across a database of non-face images, and false alarms are collected and used as the non-faces for training the subsequent strong classifier in the sequence. Training the current face-detector took about ten days on a 1.1 GHz Athlon-based PC. Fig. 12 shows the first two wavelet chosen by the system along with the tuning curves for those wavelets.

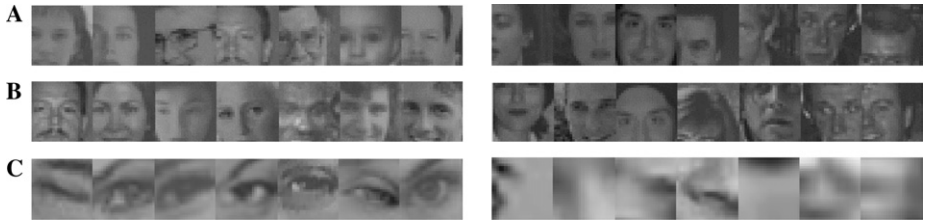

Fig. 11. Examples of positive (left) and negative (right) example patches used for training three different eye detectors. Each patch is 24×24 pixels. (A) For this detector, positive examples were chosen centered on the eye ($t = 0$), with scaling factor $q = 1$. (B) This detector uses the same scaling factor in (A), but with offset parameter t chosen such that the eye is off center to maximize pixels generated by face. (C) With a smaller value of $q = .22$, the eye fills the window.

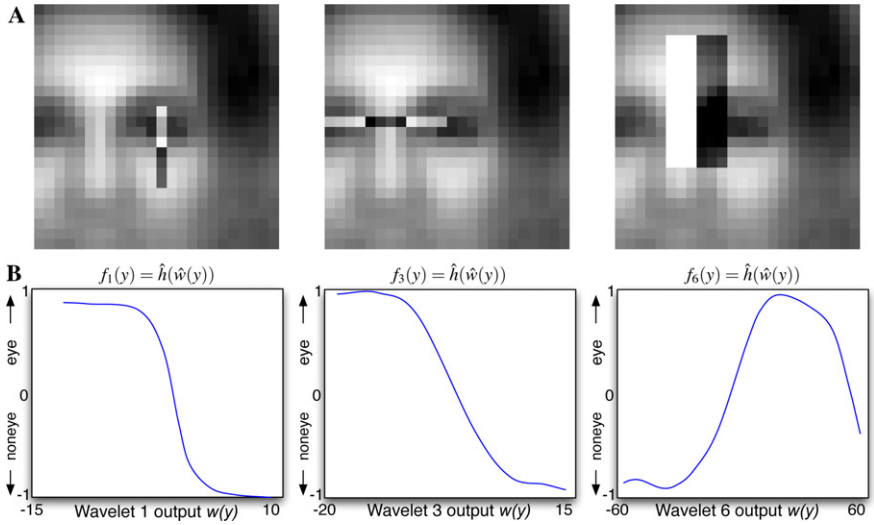

Fig. 12. The first, third, and sixth wavelets (A) and their respective tuning curves (B) for the left eye detector centered on the eye with scale factor $q = 1$. Each wavelet is shown over the average positive (eye) example. The tuning curves show the evidence for eye (high) vs. non-eye (low) as the wavelet output increases (shown increasing from left to right). The first tuning curve shows that a dark vertical region over a bright vertical region in the center of the window is evidence for an eye, and for non-eye otherwise. The middle tuning curve looks for a horizontal band that goes dark–light–dark towards the left of the window as evidence for an eye, which appears to be testing for the bridge of the nose. The rightmost wavelet also can be interpreted as a bridge of the nose detector, however it also indicates that *too much* difference between the left and right parts of the wavelet are evidence *against* eye.

At recognition time the inference algorithm calls for scanning the entire image plane and looking for square patches of arbitrary scale and location with large likelihood-ratios. In practice we start scanning patches of size 24×24, the minimum scale of interest and shift one pixel at a time until all possible patches of this size are scanned. Each larger scale is chosen to be $1.2\times$ the previous scale, and the corresponding offsets are scaled by the same proportion, for an additional $(n-24s) \times (m-24s)/s^2$ patches per scale. For a 640×480 pixel image, this produces over 400,000 total patches (see Fig. 6).

Because the early layers in the cascade need very few wavelets to achieve good performance (the first stage can reject 60% of the non-faces using only 2 wavelets, using only 20 simple operations), the average number of wavelets that need to be evaluated for each window is very small, making the overall system very fast while still maintaining high accuracy. The current system is capable of achieving 30 fps on images of 320×240 on a 3 GHz Intel Pentium 4-based desktop PC, with a minimum face size of about 24×24 pixels. Performance on the CMU-MIT dataset (a standard, public data set for benchmarking frontal face detection systems) is comparable to other state-of-the art systems. Using CMU-MIT as a validation set, we fixed performance at 92% hit rate with 10 false alarms for the experiments in this paper. While CMU-MIT contains wide variability in images due to illumination, occlusions, shadows, and differences in image quality, the performance in controlled environments, such as in the BioID dataset (used later in this study), containing faces that are frontal, focused and well lit, with simple background, is often close to 100% hit rate or frontal faces with few, if any, false alarms. While performance falls off as the face deviates from frontal (see Section 6.2), there are a wide variety of applications, in particular those in which the subject is watching a screen or driving on a road for example, for which frontal-view accuracy is sufficient. We discuss the ways to overcome this limitation in Section 7. Source code for this is stage available at http://kolmogorov.sourceforge.net.

5.2. Stage II: eye detection in the context of faces

The first stage in the eye detection system specialized on finding general regions of the image plane that are highly likely to contain eyes. The output of the system is very resistant to false alarms but does not specify well the precise location of the eyes. The second stage specializes on achieving high accuracy provided it operates on the regions selected by the previous stage. This stage uses the same searching techniques as the previous stage: all patches at multiple scales, within a sub-region of the face restricted in both location and scale, are submitted to a boosted classifier which returns the eye versus non-eye log-likelihood ratio. This log-likelihood ratio is then combined with the prior for probability of eye given location and size with respect to the face detection window to produce a final log posterior ratio of eye versus non-eye.

The data used for training was from the CMU-MIT face database and the Compaq face database used for training the face detection system. These images varied widely in image quality, lighting condition, background, facial expression, head size

and orientation, head size (with respect to the image), and image quality, and contain faces with eyes closed as well as open. Positive examples were selected by cropping patches from each image such they contain eyes at a canonical scale and location with respect to their face (described below), then scaling the patch to 24×24 pixels. Non-eye examples were taken from the same images at multiple non-eye locations and scales within the faces, with constraints described below. This resulted in 4826 positive eye examples and 10,000 non-eye examples.

There are many possible ways to crop and center the eye patches for training. We present experimental results of several different choices of cropping and centering. We can parameterize the choice by introducing variables d is the distance between the eyes, r is the ratio of the distance between the center of the eye and the left and upper edges of the face cropping window, t is an offset parameter, and q is a scale parameter. Positive training samples were then prepared by cropping example images such that $r = q(d + td)$ and scaling them to 24×24 pixels. In other words, the size of the window was chosen to be proportional to the distance between the eyes, and could be off center by some fixed amount. Thus, a small q results in a small receptive field with high resolution and a large q results in a large receptive field with relatively low resolution, while t shifts the location of the eye with respect to the center of the patch.

From the situational inference approach, one might expect that pixels which are generated by background contain relatively little additional information once we know we are within a face, thus we should choose a t and q that maximizes the number of pixels in the positive example patches that are generated by face—i.e., about the size of the face and centered on the center of the face (i.e., the eye is off-center slightly), so that very few background pixels enter into the window. However, given a fixed input size of 24×24, it is possible that smaller values of q, such as one that just covers the eye (resulting in higher resolution examples with less surrounding context) allow us to maximally benefit from the information in pixels generated by the eye only. We present results on varying these parameters experimentally to find the best choice of offset parameter t and scale parameter q in Section 6.

The situational inference approach also allows us to constrain how we choose non-eye examples: We model our prior belief about the eye location π as a normal distribution, with parameters for the mean and standard deviation of the true eye position and scale with respect to the window chosen by the face detector, as measured against the training set. In Fig. 10, we show the locations of eyes with respect to the size of the face detection window for some example data. Down on the vertical axis shows increasing ratio of the size of the face detection window to the distance between the eyes. When the face detector selects a small window relative to the true face size, resulting in a small detection width to eye distance ratio, the eyes tend to be far apart with respect to the detection window. When the face detector selects a large window compared to the distance between the eyes, the eyes tend to be located closer together, near the center of the detection window.

Using these statistics about the true eye positions with respect to the estimated face location, we can restrict the set of patches for searching—and thus for training—to a maximum Mahalanobis distance M from the mean location and scale of

each eye. Choosing $M = 16.27$ gives a 99.9% confidence interval for one of the patches containing the eye (see Appendix B).

Using these criteria, for each example face, we created two positive training examples (one for each eye), and six negative training examples, where the negative examples were selected randomly from the set of patches satisfying the maximum distance from the mean eye patch size and location criterion. To make best use of our data, we flipped the positive and negative examples from the right eye about the *horizontal* axis and combined them with the left eye examples to train a single left eye detector. Then this left eye detector was flipped about the *horizontal* axis to get a right eye detector. Examples of eyes and non-eyes used in training is shown in Fig. 11.

Once we have collected a set of positive and negative examples, we train this stage of the situational inference cascade with GentleBoost as described above. We found that it is possible to achieve excellent performance with only 50–100 wavelets without over-fitting, as tested on a validation set. Since this already allows the system to operate in real-time with high accuracy, we decided to keep the training time short (about 30 min) and the code simple and skip the attentional-cascade and boot-strap techniques for this level of the situational inference (though future work may use these techniques to see if they can slightly improve speed and/or accuracy). Fig. 9 shows example wavelets and their corresponding tuning curves for the best eye-detector.

While Stage I of our system (face detection) makes no assumptions about the number of faces on the image plane, the second stage (precise location of the eyes) assumes that there is one patch rendering the left eye and one patch rendering the right eye. If the goal is to maximize the probability of choosing the correct rendering patch optimal inference requires choosing the patch that maximizes the log-posterior ratio (21). However, if the goal is to minimize the expected squared distance from the eye, optimal inference asks for computing the mean of the posterior distribution. Both approaches can be seen as examples of a more general algorithm that chooses the N patches with highest log-posterior ratios and producing a weighted average of the opinions of those patches about the location of the feature of interest. In Section 6, we present accuracy results using different values of N.

5.3. Stage III: blink detection

Like face detection and eye detection, blink detection is done with a boosted classifier. In this case, the task is a binary classification task over a single patch per image, thus there is no need to perform a search across multiple patches. Instead, we use estimates of the eye locations to create a 44×22 pixel patch containing the eyes, doing scaling and rotation with simple linear interpolation. Training data was collected from 120 eye-open images and 120 eye-closed images collected from the Web by using the eye detector to label the eye locations, then cropping and rotating the region around the eyes to an upright frontal view. The dataset will be available at http://mplab.ucsd.edu. Fig. 13 shows examples of the training data collected this

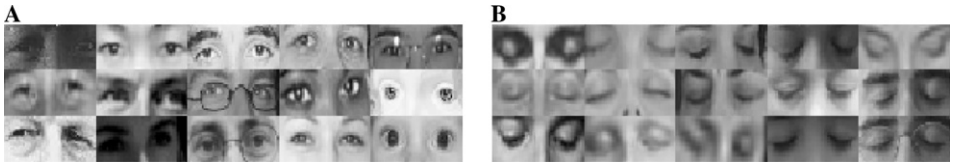

Fig. 13. Example open eyes (A) and closed eyes (B) used to train the blink detector. About 120 images of each type were taken from the web to include a wide variety of lighting conditions, facial types, glasses, and image quality. The eye detection system was used to automatically crop, scale and rotate the image patches to an upright frontal view.

way. GentleBoost is then used to select wavelets and tuning curves for this discrimination task. Fig. 14 shows example wavelets and their corresponding tuning curves for the best blink detector.

6. Experimental results

6.1. Testing datasets

We tested the performance of the eye detector on three different types of datasets. The first dataset was the BioID dataset [15,20], a freely available collection of face images with eyes labeled. This dataset contains 1521 images with good lighting conditions and frontal faces, and most subjects had their eyes open. This was to make it easier to compare our results with other eye-detection systems. The second dataset was more challenging, consisting of 400 images collected from the Web and digital cameras. We are making this dataset available at http://mplab.ucsd.edu. These images varied widely in image quality, lighting condition, background, facial expression, head orientation, head size (with respect to the image), and image quality, and contained 200 eyes-open and 200 eyes-closed examples. Measuring performance on this dataset allows us to compare how different parameter choices affect the quality of the system in unconstrained situations. We believe that if one can achieve good performance in this highly unconstrained dataset, then one can expect very good performance in better controlled situations. None of the images in this testing dataset was used in training.

The third dataset consisted of ten different heads in 153 different poses each, artificially generated from the USF Human ID 3D database [3]. Each head in the database, obtained using a laser scanner, contains structure (3D coordinates) and texture (24-bit RGB color) information for each point on the surface, suitable for rendering a high-quality still of the face at any position. Each of the ten randomly chosen heads we used for our experiments was positioned from -40 to 40° in elevation and 0–40° in azimuth, in increments of 5°, then rendered. This dataset was used to provide an estimate of the performance of the face detection and eye detection components of the system as the pose was varied.

6.2. Eye detection experiments

We tested the effect of the size and location of the receptive field used for eye detection. The receptive field size was expressed as the ratio q of the distance between the eyes. Location was expressed as "face-centered" or "eye-centered." Varying patch size from small enough to cover just the iris ($q = .11$) to large enough to cover an area four times the size of the head ($q = 2.5$) results in a U-shaped curve, with the best performance coming from the patch with size $q = 1$, which covers about 80% of the face. The best centering condition was eye-centered. The median accuracy of the best eye-detector under these conditions is 1/5 of an iris on the BioID dataset and 1/3 of an iris on the difficult dataset from the Web. Tables 2 and 3 show the results for each patch condition using different decision methods. These include choosing the maximum likelihood patch, taking the weighted average of the 10 most likely patches, taking the maximum posterior patch, and taking the weighted average of the 10 patches with the largest posterior. The fourth technique yielded the best results. Fig. 17 shows examples of this system at work.

The fact that the detector trained to consider pixels covering much of the head performs much better than the detector trained to focus on the eye-area only suggests that the detailed structure of the appearance of the eye (which at the larger resolution is mostly blurred out) is not as important as having access to the surrounding features (nose, eyebrows, corners of eyes, etc.). For the larger receptive field, dark shadows, closed versus open eyes, or specularities from glasses have less impact on the overall visual appearance of the pixels under consideration than the detector that only focuses on the eye. On the other hand, a receptive field that is much larger than the face loses the ability to discriminate much detail in the face, while considering many background pixels which have no bearing on the location of the eye within the face.

The performance on the dataset generated from the 3D database illuminates how performance changes as head-pose changes. As seen in Fig. 16, the face detector achieves about 92% for fully frontal faces (comparable to its performance on CMU-MIT), and falls off smoothly as the head deviates from frontal view. However, provided the head is detected in the first place, accuracy on eye-detection is not strongly degraded from 1/3 of an iris width as pose changes from frontal. Indeed, if elevation and azimuth is kept between $\pm 20°$, median distance from the center of the labeled eye position remains nearly constant (see Fig. 16).

6.3. Blink detection

The best performing eye detection, with scale parameter $q = 1$ and zero offset from the center of the eye, was used to automatically crop, scale, and rotate 120 examples of closed eyes and open eyes. These examples were used to train a blink detector. We stopped training after 500 wavelets and tuning curves had been chosen. The resulting classifier was then used to classify an additional 120 eyes-open and eyes-closed faces taken from the web and labeled by hand.

Table 2
Results on the BioID dataset of eye detection under different choices of patch size, offset and post-processing (mean or max of likelihood or posterior ratio)

Post-processing	$q = .11$ eye-centered	$q = .22$ eye-centered	$q = .5$ eye-centered	$q = 1$ eye-centered	$q = 1$ face-centered	$q = 1.5$ eye-centered	$q = 1.5$ face-centered	$q = 2.5$ eye-centered
Max likelihood ratio	4.66 ± 0.19	2.25 ± 0.14	0.30 ± 0.03	0.27 ± 0.01	0.41 ± 0.02	0.35 ± 0.02	0.59 ± 0.05	1.33 ± 0.06
Mean likelihood ratio	3.40 ± 0.23	2.07 ± 0.16	0.24 ± 0.04	0.21 ± 0.02	0.33 ± 0.02	0.31 ± 0.03	0.65 ± 0.04	1.26 ± 0.06
Max posterior ratio	10.43 ± 0.34	2.68 ± 0.11	0.29 ± 0.02	0.26 ± 0.01	0.41 ± 0.02	0.36 ± 0.01	0.55 ± 0.02	0.96 ± 0.03
Mean posterior ratio	9.47 ± 0.45	2.81 ± 0.16	0.24 ± 0.03	0.21 ± 0.01	0.31 ± 0.02	0.28 ± 0.02	0.55 ± 0.02	0.89 ± 0.04

Each cell displays the mean distance, in irisis, from the true center of the eye to the estimated center of the eye. The $\pm$ terms indicate standard error of the mean. The post-processing is explained in Section 5.2. The patch conditions are described in Section 6.2.

Table 3
Results on the Web dataset of eye detection under the same conditions as Table 2

Post-processing	$q = .11$ eye-centered	$q = .22$ eye-centered	$q = .5$ eye-centered	$q = 1$ eye-centered	$q = 1$ face-centered	$q = 1.5$ eye-centered	$q = 1.5$ face-centered	$q = 2.5$ eye-centered
Max likelihood ratio	4.64 ± 0.38	2.13 ± 0.19	0.38 ± 0.04	0.37 ± 0.03	0.48 ± 0.05	0.52 ± 0.05	0.67 ± 0.06	1.35 ± 0.10
Mean likelihood ratio	4.01 ± 0.46	1.82 ± 0.24	0.34 ± 0.05	0.33 ± 0.03	0.40 ± 0.05	0.47 ± 0.06	0.69 ± 0.06	1.38 ± 0.10
Max posterior ratio	6.28 ± 0.75	2.81 ± 0.23	0.38 ± 0.05	0.36 ± 0.03	0.43 ± 0.03	0.50 ± 0.04	0.60 ± 0.04	1.00 ± 0.07
Mean posterior ratio	5.78 ± 0.71	2.73 ± 0.22	0.32 ± 0.04	0.31 ± 0.02	0.36 ± 0.03	0.42 ± 0.04	0.57 ± 0.03	0.94 ± 0.06

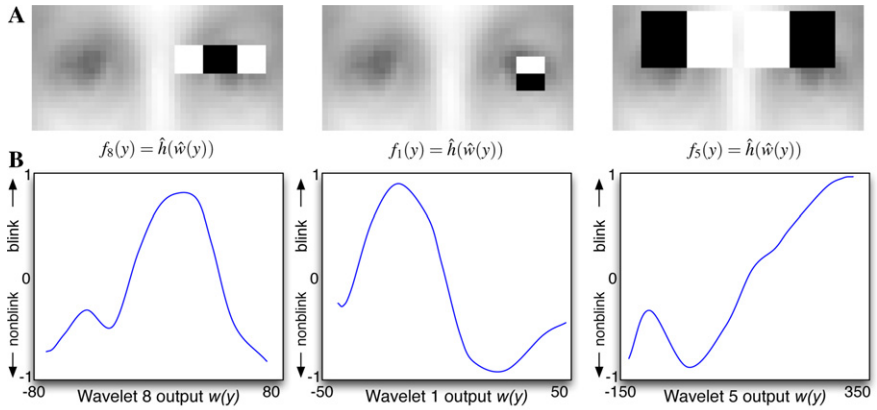

Fig. 14. Features superimposed on the average open eye image (A) and their respective tuning curves (B) for the blink detector.

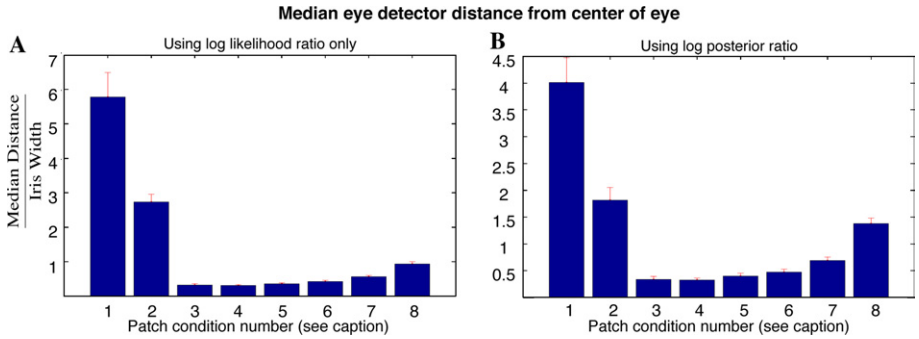

Fig. 15. Median distance from center of labeled eye positions on the Web data-set as the scale parameter q and offset parameter t are varied. The graphs show the result using only the log-likelihood ratio (A) and the log-posterior ratio, which combines the prior and likelihood (B). The conditions, described in Section 6.2, are: (1) $q = .11$, eye centered, (2) $q = .22$, eye-centered, (3) $q = .5$, eye-centered, (4) $q = 1$, eye-centered, (5) $q = 1$, face-centered, (6) $q = 1.5$, eye-centered, (7) $q = 1.5$, face-centered, and (8) $q = 2.5$, eye-centered.

To assess the effects of precise localization of the eyes we compared systems that found the eyes based on the output of Stage I alone (face detection) and systems that located the eyes using Stage I and II. The effects were dramatic: adding stage II increased performance from $56.53\% \pm 8\%$ to $83.48\% \pm 6\%$.

7. Conclusions

The study of the representations that sustain face perception in humans has recently become a subject of interest in cognitive science [5]. One heated debate centers on whether these representations are holistic in nature or whether they are are

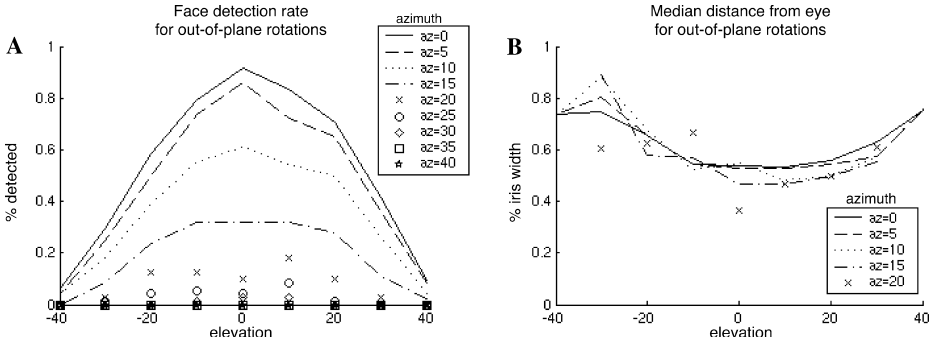

Fig. 16. Performance for face detection and eye detection as pose changes. Each curve shows performance for heads at a fixed azimuth as the elevation is varied from −40 to 40°. (A) Face detection rate falls off as pose deviates from frontal. (B) Median distance from the true eye label remains nearly constant for heads between ±20° from frontal.

feature based [9]. In line with the methodological stand of probabilistic functionalism [30] instead of positioning ourselves on this debate we focus on understanding the nature of the problem of detecting faces and facial features. To do so we developed an image generation model and derived its corresponding optimal inference algorithm. The algorithm was implemented and tested with an emphasis on robustness under natural conditions. We learned several important lessons:

(1) We found that it is very difficult to analyze eye behavior (e.g., blinks) without explicitly localizing the eyes. Based on our previous work on expression recognition we think eye localization with precision in the order of 1/4 of an iris may be necessary for reliable recognition of facial expressions. Thus it seems reasonable to expect that the brain may allocate resources to precisely locate facial features, including the eyes.
(2) We found that it is very difficult to develop detectors that are both robust (i.e., work in very general conditions) and spatially accurate. There seems to be a trade-off between robustness and accuracy. Eye detectors that localize the eyes precisely within the face exhibit unacceptable false-alarm rates when operating outside the face. Eye detectors that avoid false-alarm rates in cluttered environments, are not sufficiently precise about the location of the eyes. We explored a solution to this tradeoff, based on a cascade of detectors that operate at different levels in the robustness/localization trade-off. Some of these detectors capture the general context in which one may find eyes. By doing so they minimize false alarms at the cost of precise position information. Precise spatial localization is achieved by detectors that operate in specific contexts. If this is the strategy adopted by the brain, one would expect to find at least two types of neurons. The first type would respond to large contextual regions (e.g., faces). Neurons of this type are expected to be robust to changes in illumination but also to provide poor spatial resolution. We also expect to find a second type of neurons

Fig. 17. Examples of the eye detection system at work.

specialized on precise spatial localization of features in specific contexts. For example, neurons of this type may be maximally excited by eyes precisely aligned and maximally inhibited by small deviations from alignment. This second type of neurons may exhibit a large number of false alarms when operating out of context, making it very difficult for neuroscientists to ascertain what they respond to.

(3) In this paper, we developed the necessary likelihood-ratio and prior models using supervised learning methods. It would be of interest to investigate whether such models can be learned using unsupervised learning methods. Another possibility is that evolution took care of developing such models. Provided a set of useful wavelets is available, our face detector would require in the order of 50 kb to be encoded by the genome. It takes an additional 2 kb to encode eye detectors within faces.

(4) We focused on a system specialized on detection of eyes in a particular pose: upright frontal. In many cases (e.g., detection of fatigue in car drivers) analysis of upright-frontal views is all that is needed since frontal orientations are nominal and deviations from such orientation typically indicate fatigue or lack of attention [22]. In-plane rotation invariance can be easily achieved by scanning across rotations, in the same way we scan across scales and in-plane locations. There are several ways one could generalize the system to work under rotations in depth. One approach we experimented with in the past fits 3D morphable models and warps them into frontal views [2]. While this method is very effective under controlled illumination conditions, it is expensive computationally and brittle when exposed to outdoor conditions. Another approach we are pursuing is a mixture of experts architecture, where each expert specializes on specific face views. Indeed there is experimental evidence for the existence of view specific face detection neurons in infero-temporal cortex (IT) in monkeys [28]. Due to rotational symmetry of the face, pose invariance can be achieved by covering an octant of the sphere of possible face orientations, i.e., $\pi/2$ steradians. Assuming each pose expert can handle $\pm 5°$, as is the case on the system presented here, it would take approximately $1/(2\tan(5)) \approx 6$ experts to cover an octant. This is certainly not an unreasonable number of experts, thus making mixtures of pose experts a very attractive architecture for future systems. Development of systems specialized in non-frontal views is currently difficult due to the lack of labeled datasets that include sufficient number of images in multiple poses and illumination conditions. Collecting such databases is critical to accelerate progress in this field.

Appendix A. Examples

See Fig. 17.

Appendix B. Gaussian confidence regions

Let Z be n-d Gaussian, zero mean with covariance I_n. Let σ a covariance matrix, with eigenvectors p and eigenvalues λ, i.e., $\sigma = p\lambda p^{\mathrm{T}}$. Let $\mu \in \mathscr{R}^n$. Let $Y = p\ (\lambda)^{1/2} Z + \mu$. Thus Y is Gaussian with covariance Σ and mean μ.

For a given $\alpha > 0$ we want the probability that $(Y-\mu)^{\mathrm{T}}\ \Sigma^{-1}(Y-\mu)$ takes values smaller or equal to α. Now note

$$(Y-\mu)^{\mathrm{T}}\Sigma^{-1}(Y-\mu) = Z^{\mathrm{T}}Z = \sum_{i=1}^{n} Z_i^2, \tag{B.1}$$

which is a χ^2 random variable with n degrees of freedom. This is the key to obtaining confidence intervals.

B.1. Example

Suppose $n = 3$, Y is gaussian with mean μ and covariance σ and we want to calculate the value α such that

$$P((Y-\mu)^{\mathrm{T}}\sigma^{-1}(Y-\mu) < \alpha) = 0.001,$$

i.e., we want a volume that captures 99.9% of the probability. First we go to the χ^2 distribution with 3° of freedom and find that the critical value for 1/1000 is 16.27. Thus

$$P((Y-\mu)^{\mathrm{T}}\sigma^{-1}(Y-\mu) < 16.27) = P(Z^{\mathrm{T}}Z < 16.27) = 1/1000. \tag{B.2}$$

Thus the 99.9% confidence region for Y is given by the set of values y such that

$$(y-\mu)^{\mathrm{T}}\sigma^{-1}(y-\mu) \leqslant 16.27. \tag{B.3}$$

References

[1] S. Baron-Cohen, Mindblindness, MIT Press, Cambridge, MA, 1995.
[2] M. Stewart Bartlett, B. Braathen, G. Littlewort, E. Smith, J.R. Movellan, An approach to automatic recognition of spontaneous facial actions, in: Advances in Neural Information Processing Systems, number 15. MIT Press, Cambridge, Massachusetts, (in press).
[3] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces. in: SIGGRAPH'99 conference proceedings, pp. 187–194. 1999.
[4] J.F. Cohn, J. Xiao, T. Moriyama, Z. Ambada, T. Kanade, Automatic recognition of eye blinking in spontaneously occurring behavior, Behav. Res. Methods Instr. Comp. (in press).
[5] G.W.G.W. Cottrell, M.N. Dailey, C. Padgett, R. Adolphs, Is all face processing holistic? the view from UCSD. in: M. Wenger, J. Townsend (Eds.), Computational, Geometric, and Process Perspectives on Facial Cognition: Contexts and Challenges. Erlbaum, 2003.
[6] S. Edleman, L.M. Vaina, David marr, in: International Encyclopedia of the Social and Behavioral Sciences, 2001.
[7] P. Ekman, Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage, first ed., W.W. Norton, New York, 1985.
[8] P. Ekman, W. Friesen, Facial action coding system: a technique for the measurement of facial movement, Consulting Psychologists Press, Palo Alto, CA, 1978.
[9] M.J. Farah, K.D. Wilson, M. Drain, J.N. Tanaka, What is special about face perception?, Psychological Review 105 (3) (1988) 482–498.
[10] T. Farroni, G. Csibra, F. Simion, M.H. Johnson, Eye contact detection in humans from birth, in: Proceedings of the National Academy of Sciences, 99 (2002) 9602–9605.
[11] I.R. Fasel, E. Smith, M.R. Bartlett, J.R. Movellan, A comparison of Gabor filter methods for automatic detection of facial landmarks. in: Proceedings of the 7th Symposium on Neural Computation. California Institute of Technology, 2000.

[12] Y. Freund, R. Schapire, A short introduction to boosting, J. Japan. Soc. for Artif. Intel. 14 (5) (1999) 771–780.
[13] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: International Conference on Machine Learning, pp. 148–156, 1996.
[14] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, 1998.
[15] R. Frischholz, U. Dieckmann, BioID: A multimodal biometric identification system, IEEE Computer 33 (2) (2000).
[16] N. George, J. Driver, R.J. Dolan, Seen gaze-direction modulates fusiform activity and its coupling with other brain areas during face processing, Neuroimage 6 (13) (2001) 1102–1112.
[17] A. Haro, M. Flickner, I.A. Essa (Eds.), Detecting and Tracking Eyes by Using Their Physiological Properties, Dynamics, and Appearance, IEEE Computer Society, 2000, ISBN 0-7695-0662-3.
[18] M.K. Holland, G. Tarlow, Blinking and mental load, Psychological Reports (31) (1972) 119–127.
[19] J. Huang, H. Wechsler, Eye detection using optimal wavelet packets and radial basis functions (RBFs), International Journal of Pattern Recognition and Artificial Intelligence 7 (13) (1999).
[20] O. Jesorsky, K. Kirchberg, R. Frischholz, Robust face detection using the hausdorff distance, in: J. Bigun, F. Smeraldi (Eds.), Audio and Video based Person Authentication, Springer, 2001, pp. 90–95.
[21] Q. Ji, X. Yang, Real time visual cues extraction for monitoring driver vigilance. in: Second International Workshop on Computer Vision Systems (ICVS2001), 2001.
[22] Q. Ji, X. Yang, Real-time eye, gaze, and face pose tracking for monitoring driver vigilance, Real-Time Imaging 3 (8) (2002) 1077–2014.
[23] M.H. Johnson, The developmental and neural basis of face recognition: Comment and speculation, Infant and Child Development 10 (2001) 31–33.
[24] C.N. Karson, Physiology of normal and abnormal blinking, Advances in Neurology 25-37 (49) (1988) 119–127.
[25] R. Kawashima, M. Sugiura, T. Kato, A. Nakamura, K. Hatano, K. Ito, H. Fukuda, S. Kojima, K. Nakamura, The human amygdala plays an important role in gaze monitoring: A PET study, Brain 122 (4) (1999) 779–783.
[26] R. Kothari, J. Mitchell, Detection of eye locations in unconstrained visual images. ICIP96, 1996.
[27] T. Leung, M. Burl, P. Perona, Finding faces in cluttered scenes using random labeled graph matching, in: 5th International Conference on Computer Vision, 1995.
[28] N.K. Logothetis, T. Poggio, Viewer-centered object recognition in monkeys. Technical Report A.I. Memo 1473, Artificial Intelligence Laboratory, M.I.T., 1994.
[29] D. Marr, Vision, Freeman, New York, 1982.
[30] J.R. Movellan, J. Nelson, Probabilistic functionalism: A unifying paradigm for the cognitive sciences, Behavioral and Brain Sciences 24 (4) (2001).
[31] J. Phillips, DARPA Symposium on Human ID, Washington, DC, 2003.
[32] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. on Pattern Analysis and Machine Intelligence 1 (20) (1998) 23–28.
[33] G. Shakhnarovich, P.A. Viola, B. Moghaddam, A unified learning framework for real-time face detection and classification, in: International Conference on Automatic Face and Gesture Recognition, 2002.
[34] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman and Hall, 1986.
[35] K.K. Sung, T. Poggio, Example based learning for view-based human face detection, IEEE Trans. Pattern Anal. Mach. Intelligence 20 (1998) 39–51.
[36] K. Van-Orden, T.P. Jung, S. Makeig, Eye activity correlates of fatigue, Biol. Psychol. 3 (52) (2000) 221–240.
[37] P. Viola, M. Jones, Robust real-time object detection. Technical Report CRL 20001/01, Cambridge Research Laboratory, 2001.
[38] L. Wiskott, J.M. Fellous, N. Krüger, C. von der Malsburg, Face recognition by elastic bunch graph matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 775–779.