Towards Practical Facial Feature Detection

Micah Eckhardt

Institute of Neural Computation
University of California, San Diego
La Jolla, CA 92093

micahrye@mplab.ucsd.edu

Ian Fasel

Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712

ianfasel@cs.utexas.edu

Javier Movellan

Institute of Neural Computation
University of California, San Diego
La Jolla, Ca 92093

movellan@mplab.ucsd.edu

Localizing facial features is a critical component in computer vision applications such as person identification and expression recognition. Practical applications require detectors that operate reliably under a wide range of conditions, including variations in illumination, ethnicity, gender, age, and imaging hardware. One challenge for the development of such detectors is the inherent tradeoff between robustness and precision. Robust detectors provide poor localization and detectors sensitive to small shifts, which are needed for precise localization, generate a large number of false alarms. Here we present an approach to this tradeoff based on context dependent inference. First robust detectors are used to detect contexts in which target features occur, then precise detectors are trained to localize the features given the context. This paper describes the approach and presents a thorough empirical examination of the parameters needed to achieve practical levels of performance, including the size of the training database, size of the detector's receptive fields, and methods for information integration. The approach operates in real time and achieves, to our knowledge, the best performance to-date reported in the literature.

Keywords: machine vision; feature detection; face recognition

1. Introduction

Locating facial feature points (FFPs) is a critical component in computer vision applications, such as person identification and facial expression analysis [6, 7, 4, 11, 19]. Despite its importance FFP localization is still an unsolved problem for applications that need to operate under a wide range of conditions, including variations in illumination, ethnicity, gender, age, pose, and imaging hardware [4]. One challenge for

the development of such detectors is an inherent tradeoff between robustness and precision. Robust detectors provide poor localization performance and detectors capable of distinguishing small deviations from target locations tend to generate a large number of false alarms. Here we present an approach to this tradeoff based on context dependent inference. First robust detectors are trained to detect the context in which target features occur and then precise detectors are trained to localize the target features given the context.

After presenting the proposed context dependent inference (CDI) architecture, we provide a thorough exploration of the parameters needed to achieve practical performance levels, including the size and character of the training database, size of the detector's receptive field, feature search location, strong classifier composition and methods for information integration. This careful analysis of the factors affecting performance is useful not only for understanding the proposed CDI architecture, but also for many more general issues which arise in facial feature detection and face analysis.

2. Overview of the Approach

A general approach to feature detection, based on the idea of context dependent inference, was presented in [7]. While here we maintain the same theoretical framework, we use larger training databases and present a detailed analysis of the different factors affecting performance. The results is a system with unsurpassed levels of performance in literature published to date.

There is a fundamental tradeoff inherent to the problem of feature localization: robust feature detectors tend to localize poorly and detectors sensitive to small variations, needed for precise localization, tend to produce a large number of false alarms. One common approach to solve this tradeoff is based on the operation of a set of independent feature detectors [8, 10]. 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.) are integrated by considering spatial configurations that match the distribution of inter-feature distances typical of the human face [12, 13, 21, 5]. Unfortunately the computational complexity of this approach scales exponentially with the number of false alarms of each feature detector, and the number of basic feature detector types.

The approach we propose here is based on context dependent inference, an idea that was first formalized by Yuille and Bultzoff [23] to help explain biological vision. They proposed that it is too difficult to develop context independent perceptual systems capable of operating robustly and precisely under all possible conditions. Instead they proposed that perceptual inference may be better handled using context-dependent experts, each specialized on making inferences given a specific context. The essence of this idea can be formalized as follows: let y be an observed image, t the location of a target feature (e.g., the right eye) and c the image region rendering a context relevant to this target (e.g., the set of pixel loca-

tions in the image that render human faces). Our goal is to infer the location of the target feature on the image plane. The information needed to solve this problem is contained in the posterior probability of the target t given the image y. Using the law of total probability we have that

$$p(t \mid y) = \sum_c p(c \mid y) p(t \mid c, y) \tag{1}$$

where $p(c \mid y)$ is the posterior probability of a context given the observed image. For example c may partition the image into pixels rendering a face and pixels rendering a generic everything else. The term $p(t \mid c, y)$ is a context specific target detector. It provides information about the location of the target, provided it operates in a specific context c. Thus (1) tells us that if we want to localize a target feature we can do so by combining the output of a system that detects the relevant context in which the targets occur and another system that localizes the target in given contexts.

2.1. *Real-Time Inference Architecture*

As described above the first component of the inference process locates relevant contexts and the second makes inferences within these contexts. In our case we only use two contexts: faces, and background. We also assume the target features only occur in the context of faces. Thus, under these assumptions we just need to develop two modules: (1) a face detector and (2) target feature detectors trained to operate in the context of faces.

Here we investigate context based feature detectors, which are based on the mustiscale "sliding window" approach first popularized by Rowley et al. [16] for the task of face detection, and later made particularly efficient by Viola and Jones [18]: Binary classifiers are trained on a set of image patches of fixed size (in our experiments we use 24×24 pixel image patches). At run time an image is rapidly scanned at multiple locations, and the detector makes binary decisions for each of the scanned patches. Faces larger than the original size are found by repeating the search in copies of the image scaled to smaller sizes. Thus, a 24×24 pixel face in a $1/4$ size copy of the image corresponds to an 96×96 pixel face in the corresponding location in the original.

Figure 1 describes the general workflow of the inference architecture. It consists of two stages: the first stage detects the context under which the target features occur, in this case human faces. This stage operates under very general background and illumination conditions narrowing down the plausible locations of the target features on the image plane. It is based on the Viola-Jones architecture, but with two key changes: (1) We use continuous non-parametric transfer functions rather than binary threshold functions. This allows us to obtain continuous likelihood ratio estimates for each possible image patch, rather than just binary decisions. (2) We eliminate the cascaded architecture in favor of a probabilistic sequential decision making architecture. A full description of this module can be found in [7].

The second stage, which is the focus of this paper, specializes in achieving high localization accuracy of the target features, provided it operates on the regions selected by the previous stage. This second stage uses the same inference architecture as the previous stage, but now applied only to the detected face region. The combined system operates in real time at video frame reates (e.g., 30 frames per second). However, it treats each frame as independent of the previous frames, allowing for the application of feature localization in video or still images.

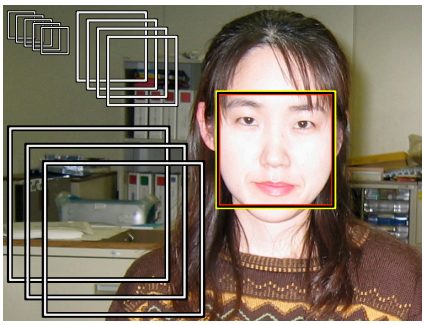
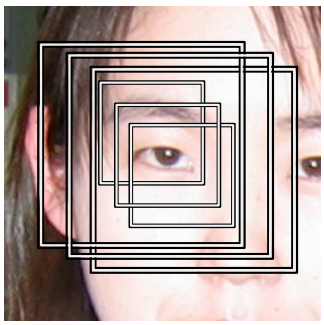
a)  b) 

Fig. 1. *Context sensitive search (a) First the contexts of interest are detected, in this case faces. (b) Next the target features are precisely localized given the context. In both cases we use a multi-scale sliding-window detector approach.*

2.2. *Learning Architecture*

The proposed approach requires conditional likelihood-ratio estimates, i.e., given an arbitrary image patch y we need an estimate for the ratio between the probability of such a patch being generated by the target class vs. the background class. Here we learn these likelihood ratios using a *boosting* algorithm known as *GentleBoost* [9]. Boosting [**?**] refers to a family of machine learning algorithms that builds accurate (strong) classifiers by combining a collection of weak classifiers. Each of these weak classifiers is chosen in a sequential manner for its capacity to reduce the mistakes made by the current collection of weak classifiers. While each weak classifier may perform only slightly above chance, the combined system (i..e, the strong classifier) may achieve very high levels of accuracy. In [9] it was shown that boosting methods can be reinterpreted from the point of view of sequential maximum likelihood estimation, an interpretation that makes it possible to use thesse methods within the framework proposed here.

Learning in GentleBoost is accomplished by sequentially choosing weak classifiers and combining them to minimize a chi-square error function. In our application, each weak classifier consists of a simple linear filter (selected from a large, fixed library of filters), followed by a non-linear transfer function. The pool of filters

we use are the same as used in [18], with the addition of a center-surround filter class (see Figure 2). The main reason for using these relatively simple features is that they can be computed very efficiently in general purpose computers without the need of specialized hardware. (See Viola and Jones [18], Shakhnarovich et al. [17] for a more detailed explanation). In [18], the nonlinear transfer function was a simple threshold function whose output was in the set $\{-1, +1\}$. In this paper, we use a piecewise constant function whose parameters are chosen by the GentleBoost algorithm. This allows each weak classifier to output arbitrary real values in the range $[-1, +1]$ rather than simply binary decisions.

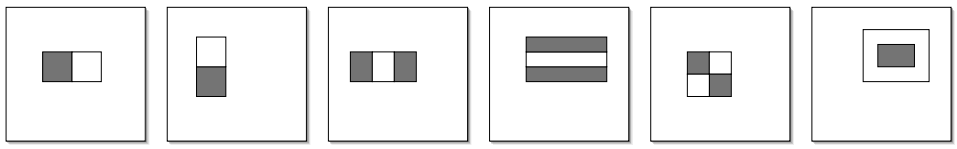


Fig. 2. *Each filter is computed by taking the difference of the sums of the pixels in the white boxes and grey boxes. Filter types include those in [18], plus a center-surround type filter.*

3. Description of databases

The following is a short description of the most commonly used databases for training and testing facial feature detectors. FERET-frontal [15] is a free, publicly available database with 3880 images taken in controlled settings with no background clutter and little variation in illumination. XM2VTS [22] and BANCA-C/WorldModel [1] are commercially available databases. XM2VTS contains 1180 frontal face images and BANCA-C/WorldModel contains 2380 frontal face images. Both XM2VTS and BANCA-C/WorldModel are similar to the FERET-frontal database. These databases are not considered representative of real-world conditions and lack difficult localization elements.

The BANACA-D/A and BioID [2] databases attempt to simulate real-world conditions. BANACA-D/A contains 4160 frontal face images with cluttered backgrounds, variable illuminations and slight head pose variation. The free and publicly available BioID database contains 1521 frontal face images that vary with respect to illumination, background, face size and slight head pose variation. Based on results from literature [7, 11, 24], these databases are considered more challenging.

While these databases have helped to advance research in the area of facial feature detection and face recognition, they are not large enough and are not sufficiently representative of the imaging conditions likely to be encountered in practical "everyday" applications (e.g., consumer cameras, or social robots). For this reason we collected a new database, named GENKI, that currently contains 70,000 images collected from the World Wide Web (See Figure 3). This collection contains

Fig. 3. Sample face images from the GENKI database.

a very wide variety of different imaging conditions with respect to illumination, background clutter, head pose, age, ethnicity, partial occlusions, image compression artifacts, and imaging devices. Images in the database were hand labeled for the location of the temporal and nasal corner of right and left eyes, the center of the tip of the nose, mouth center (defined as the estimate of the location of the intersection between the line defined by the labial furrow and the curve defined by the end of the upper teeth) and pose: roll; pitch and yaw. Head pose ranges are: Pitch -30^o, +48^o; Yaw +/- 62^o; Roll -60^o, +53^o. For our purposes the center of the eye is defined as the midpoint between the labeled temporal and nasal corner of the eye.

4. Empirical Studies

We present a set of experiments whose goal is to demonstrate the effects of several implementation parameters on system performance, and identify trends for improving performance. The set of all possible parameter settings is too large to search exhaustively, therefore we proceeded in a sequential manner, identifying promising parameter values, fixing them and then varying other parameters. Parameter values which remain fixed throughout the work are based on preliminary tests and previous work [7]. We present performance on three types of facial features: (1) the center of the eyes (here we only report right eye performance, since left eye performance is equivalent); (2) tip of the nose, and (3) center of the mouth (See Figure 4).

Although not consistently so, the trend in the literature is to report performance on facial feature localization in terms of inter-ocular distance [3, 5, 11], i.e. the distance between the centers of the eyes. This unit of measurement can be difficult to interpret intuitively, therefore we have proposed an alternative *standard iris*

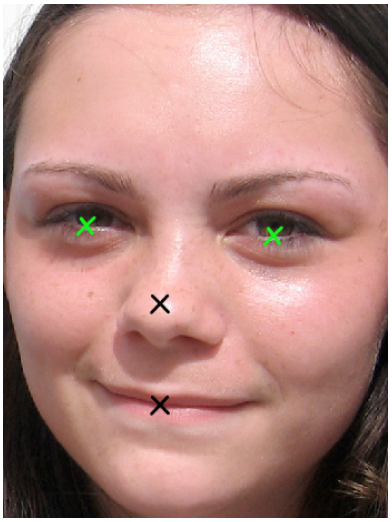


Fig. 4. Face with hand labeled features.

widths unit of measurement [7]. A standard iris width is 1/7 of the inter-ocular distance, and represents the average ratio of the width of the iris versus the inter-ocular distance. The value of 7 standard iris widths per interocular distance is based on statistical analysis of a sample from the GENKI database. In that sample, on average the interocular distance was 6.63 times larger than the diameter of the iris. For simplicity we chose the closest integer to 6.63 as a standard.

When comparing the output of the detector to the hand labeled feature locations, the "error" of a specific feature detection is defined as the euclidean distance (in standard iris widths) between the system's final output and the hand labeled location. Here we report two error statistics: the root mean square error (RMSE), and the median absolute error (MAE), which is less sensitive to outliers. Another useful measure is the proportion of times that the absolute distance between the system's output and the hand labels is below a threshold. Unless otherwise stated, we use the relatively lax threshold of 1.75 standard iris widths.

In the remainder of this section we report on a series of experiments designed to investigate the effects of the following factors: (1) size of the search region, (2) receptive field size, (3) sampling distance (4) training set size, (5) selection of positive and negative examples, (6) number of rounds of training, (7) information integration, and (8) pose variation.

4.1. *Size of the Search Region*

As described in section 2.1 and illustrated in Figure 1, we use a sliding-window approach to searching for target facial features at different locations and scales. First a context detector segments a region of the image likely to contain a face

(which should contain the target features). In this initial face detection stage, we first scan 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 approximately 1.2 times the previous scale, and the corresponding offsets between windows are scaled by the same proportion.

After the face is found, we search for facial feature points (FFPs) within the face. Rather than search within the entire face region for a particular FFP, we limit our search to the areas that will most probably contain the desired feature. We model the joint prior distribution of the offset (in horizontal and vertical directions) and scale as a three dimensional Gaussian distribution. The parameters for this distribution are simply the empirical mean and covariance of these values found in a sample from the GENKI database. The resulting model was used to determine search regions for the contextual feature detectors. We refer to these search region as *regions of interest*, or *ROI*s. The size of a ROI is determined by a maximum allowed Mahalanobis distance from the most probable prior location of the target features. We tested the effect of varying this maximum threshold over the following Mahalanobis distances: $\{1.215, 2.366, 4.108, 6.251, 16.275, 21.101, 25.902, 30.665\}$. As shown in Figure 5 performance increased with larger search regions, but improvements were very small for search regions with Mahalanobis distances beyond 15. Therefore in the remaining experiments we fixed the ROI threshold distance to 15.

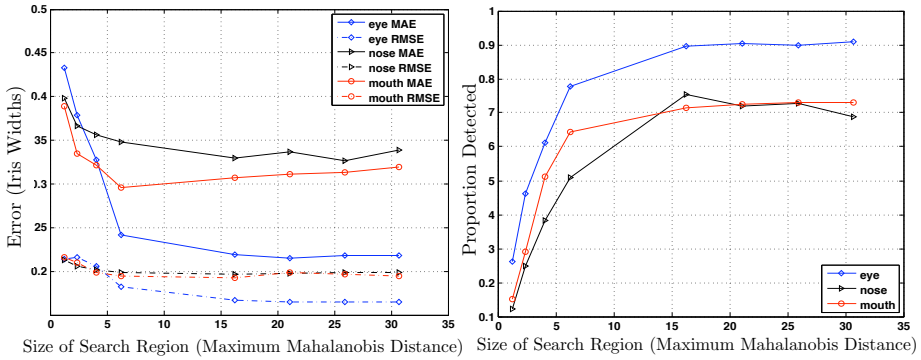


Fig. 5. Effect of Search Region Size.

4.2. *Receptive Field Size*

The size and resolution of the detector (See Fig. 6) has a strong influence on the localization performance and it is unclear apriori what the optimal values should be: For example, should eye detectors search only for the pixels making up the iris, should they use information from the entire eye region, or should they use

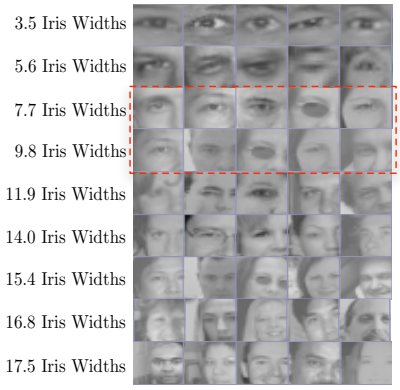


Fig. 6. Examples of varying receptive field size (Optimal sizes in dashed box).

information far beyond the eye region? To study this issue we tested the effect of varying the size of the detector's *receptive field*, defined as a square image patch of fixed width relative to the size of the detected face. The tested receptive field widths, measured in standard iris widths, were $\{3.5, 5.6, 7.7, 9.8, 11.9, 14.0, 15.4, 16.8, 17.5\}$ (See Fig. 7).

While the amount of face context varied within the receptive field, the size of the training patches was always scaled to 24×24 pixels. Thus in our approach there is a context versus resolution tradeoff – larger receptive fields mean less resolution, while smaller receptive fields mean greater resolution.

Best performance was obtained by using rather large receptive fields that include a significant amount of the rest of the face (See Fig. 7). This result, which is somewhat unintutivie, replicates our previous results on a smaller dataset [7]. It should be pointed out that other approaches typically use much smaller receptive field sizes, typically about 3 iris widths [5, 3, 14, 19].

4.3. *Sampling Rate*

Because of the sliding-window approach, performance is influenced by the distance between each application of the detector window. We refer to this distance as s for *sampling distance* (i.e. inverse of the sampling rate).

Given a *ROI* for a particular target feature, we first choose the minimum $K \times K$ scale allowed by the *ROI* and apply the classification window at every point on a grid within the *ROI* whose nodes are spaced every $s = \frac{K}{24}$ pixels, rounded to the nearest integer. K is then incremented to the next positive integer multiple of 24 and the process is repeated, until K is larger than the maximum specified by the *ROI*. In practice, the output of the feature detectors is sensitive to translations and scales smaller than this base resolution. Since the true location of a FFP may be between

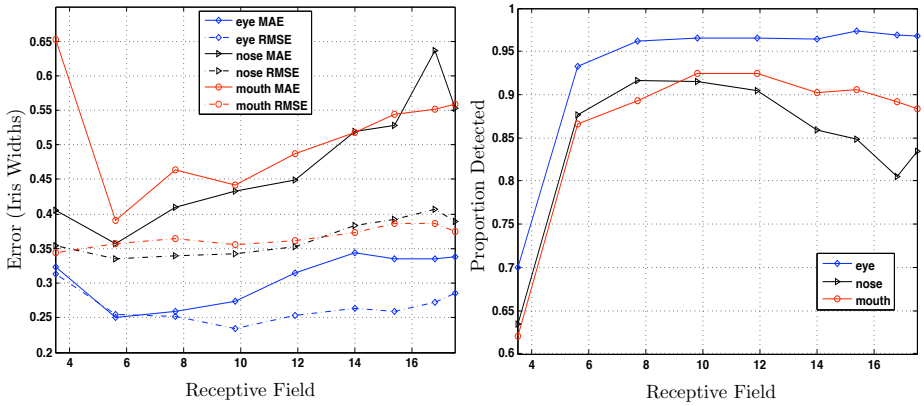


Fig. 7. Effect of Receptive field size.

Table 1. Performance as a function of sampling rate

Feature	**MAE**		**RMSE**		**detection rate**	
	$\frac{s}{2}$	s	$\frac{s}{2}$	s	$\frac{s}{2}$	s
Right eye	0.2268	0.2808	0.2406	0.2696	0.9645	0.9576
Left eye	0.2290	0.2780	0.2338	0.2573	0.9644	0.9623
Nose	0.3689	0.3977	0.3164	0.3341	0.9185	0.9093
Mouth	0.3391	0.5258	0.3311	0.3849	0.8992	0.8638

the regions scaneed by the sliding window, we tried increasing the sampling rate by a factor of 2, i.e., for a particular scale K, choose windows spaced $\frac{s}{2}$ pixels apart. Table 1 shows the effect of increasing the sampling rate: Doubling the sampling rate reduced error rates by a few percentage points for the nose and mouth detectors. However, the eye detectors improved by less than one percent. Throughout the remainder of experiments we sampled every $\frac{s}{2}$ pixels.

4.4. *Training Set Size*

We investigated the effects of the training set on performance. The number of positive examples was varied from 100, to 20000 according to the following schedule : $\{100, 300, 500, 1000, 2000, 3000, 5000, 8000, 10000, 15000, 20000\}$. For each condition the number of negative examples was 3 times the number of positive examples.

Results are presented in Figure 8. While there are some performance gains for training sizes beyond 5000 examples, they are very small. Across all feature types, training set sizes between 1000 and 5000 examples perform well. This was a surprisingly small number considering the improvements made in other problems,

such as face detection and smile recognition, by increasing the number of training examples beyond 10,000 [18, 20]. This suggests that restricting the context to the region of the face near the features reduced the complexity of the problem, i.e., detecting eyes in frontal faces is an easier problem than detecting frontal faces in general background conditions.

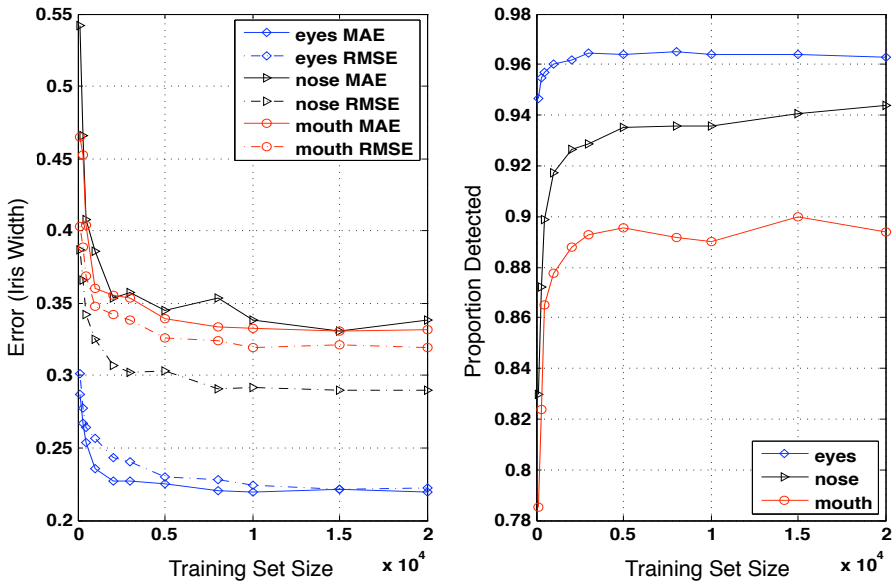


Fig. 8. Effect of training example set size.

4.5. *Selection of Positive and Negative Examples*

The training patches were chosen using the sliding-window described in Section 4.3. For each image in the training set, for each particular feature, we first create a "candidate list" of all locations visited by the detection window, restricted to the region of interest (shown by the green dashed outer ellipse in Fig. 10). Thus the scanning procedure to select training patches is identical to the scanning procedure used at runtime to detect target features. Once we have a collection of candidate patches we select the patch with minimum Euclidian distance from the human labeled feature point (yellow X in Fig. 10), and add it to the set of positive training patches. This ensures that the system is trained with positive examples that have the FFP centered and slightly shifted from center with respect to location and scale in a manner similar to how they will be encountered at runtime. This is

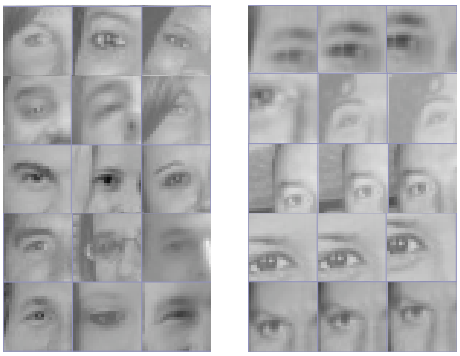


Fig. 9. Training Examples for receptive field of size 7.7 iris widths. Exclusion range of 2 with a bias selection to patches nearer the *ROI* boundary: A) Positive training examples, B) Negative training Examples.

particularly important since patches visited in the sliding-window approach are sometimes separated by several pixels each depending on the size of the face.

To select negative examples, we first create an exclusion region (white inner eclipse in Fig. 10) around the labeled feature location and take a sample of patches inside the *ROI* but outside of the exclusion region. In addition we biased the sampling process in three ways: (a) no bias, (b) bias towards selection of patches close to the exclusion boundary, and (c) bias towards selection of patches nearer to the *ROI* boundary. Experimental results indicate method (c) in conjunction with a moderate exclusion range (Fig. 10 examples C and D) gave the highest performance.

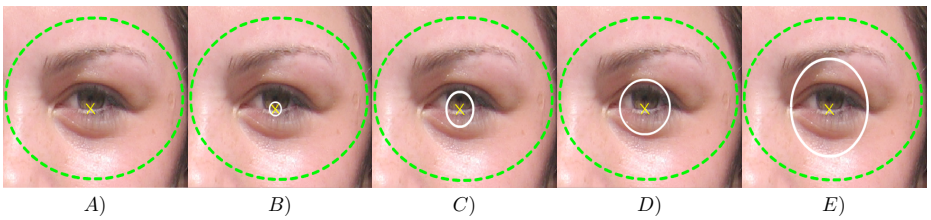


Fig. 10. Left eye region of interest: A) No exclusion region, B) Exclusion region of 0.5, C) Exclusion region of 1, D) Exclusion region of 2 and E) Exclusion region 3

4.6. *Number of Training Rounds*

Contrary to other machine learning methods, boosting is a sequential procedure in which a new filter is added after each round of training. The amount of training, i.e.,

the number of filters chosen to construct a classifier, influences training time, runtime, detection rate and localization accuracy. We investigated the effect of varying the number of training rounds on classification performance. Experimental results (See Fig. 11) indicate optimal levels of performance between 30 and 150 training rounds, with only nominal improvements in performance beyond that. Other problems like face detection typically require several thousand training rounds [18] indicating that feature detection in the context of faces is a less complex problem. Based on these results we limited the rounds of training to 150 for all subsequent experiments.

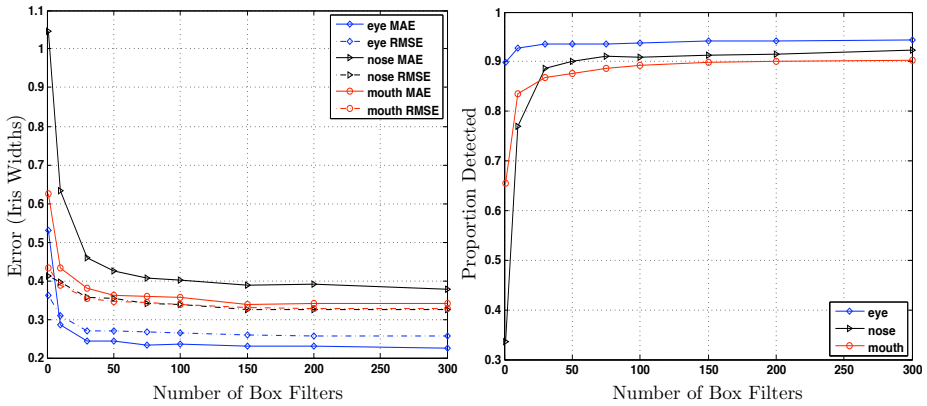


Fig. 11. Performance with respect to the number of box filters choosen to construct the strong classifier.

4.7. *Information Integration*

For each image patch visited by the multi-scale sliding window approach, the detector returns the log-likelihood ratio that the patch was generated by the target class vs. the background class. The result is a list of candidate patches that can be assessed in terms of the likelihood that they are of the target class. This can then be combined with the prior probabilities that the feature is located at a particular location given the face detection region to form a posterior probability estimate at every location.

We investigated several methods of integrating the results over this list of likelihoods and posterior probability estimates to make a final decision about the FFP location: (1) choosing the maximum log-likelihood patch; (2) choosing the maximum log-posterior patch; (3) choosing the mean of the 23 highest log-likelihood patches; (4) choosing the mean of the 23 highest log-posterior patches; (5) choosing

the median of the 23 highest log-likelihood patches; (6) choosing the median of the 23 highest log-posterior patches; (7) choosing the weighted average of all log-likelihood patches; (8) choosing the weighted average of all log-posterior patches. The use of 23 patches was determined during preliminary experiments.

Figure 12 shows the results. The eyes and nose perform best by taking the mean of the top 23 highest log-likelihood patches (method (3) above). Regardless of feature type, there is a similar trend between integration methods (though the eyes consistently out perform the nose). One exception is mouth detection which shows inconsistent results depending on the statistic used to measure performance. Based on our results we use method (3) for eyes and nose information integration and method (4) for mouth information integration for the remaining experiments.

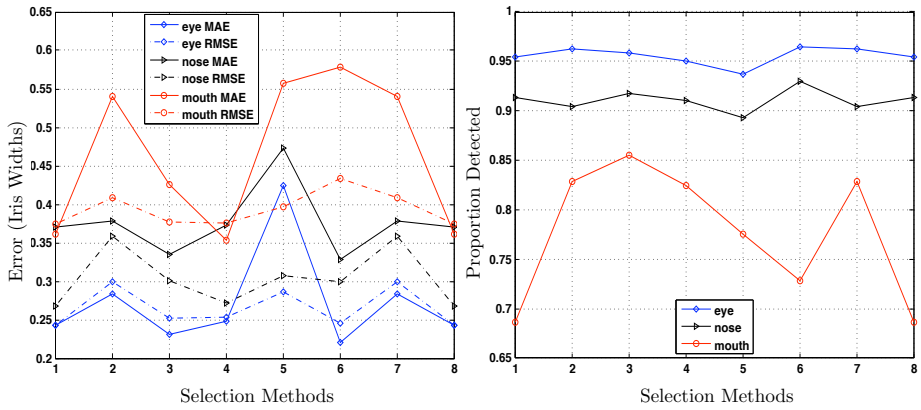


Fig. 12. Information integration methods: (1) maximum log-likelihood; (2) maximum log-posterior; (3) mean of the 23 highest log-likelihoods; (4) mean of the 23 highest log-posteriors; (5) median of the 23 highest log-likelihoods; (6) median of the 23 highest log-posteriors; (7) weighted average of all log-likelihoods; (8) weighted average of all log-posteriors

4.8. *Pose Invariance*

Figure 13 shows the performance of individual detectors, as well as the face detector, as a function of the head pose (roll, pitch and yaw). The ranges, in terms of distance from frontal, are $(-5°, 5°)$, $(\pm 5°, \pm 15°]$ and $(\pm 15°, \pm 60°]$. These results are based on 150 images per pose range randomly selected from the GENKI dataset. The figure shows that face detection was somewhat fragile beyond $\pm -15°$, with respect to roll and pitch. Detection of the tip of the nose and mouth were quite sensitive to pose variations in roll and yaw. However eye detection proved to be quite robust to pose variation and could handle relatively well pose variations up $\pm 60°$ in roll, yaw and pitch.

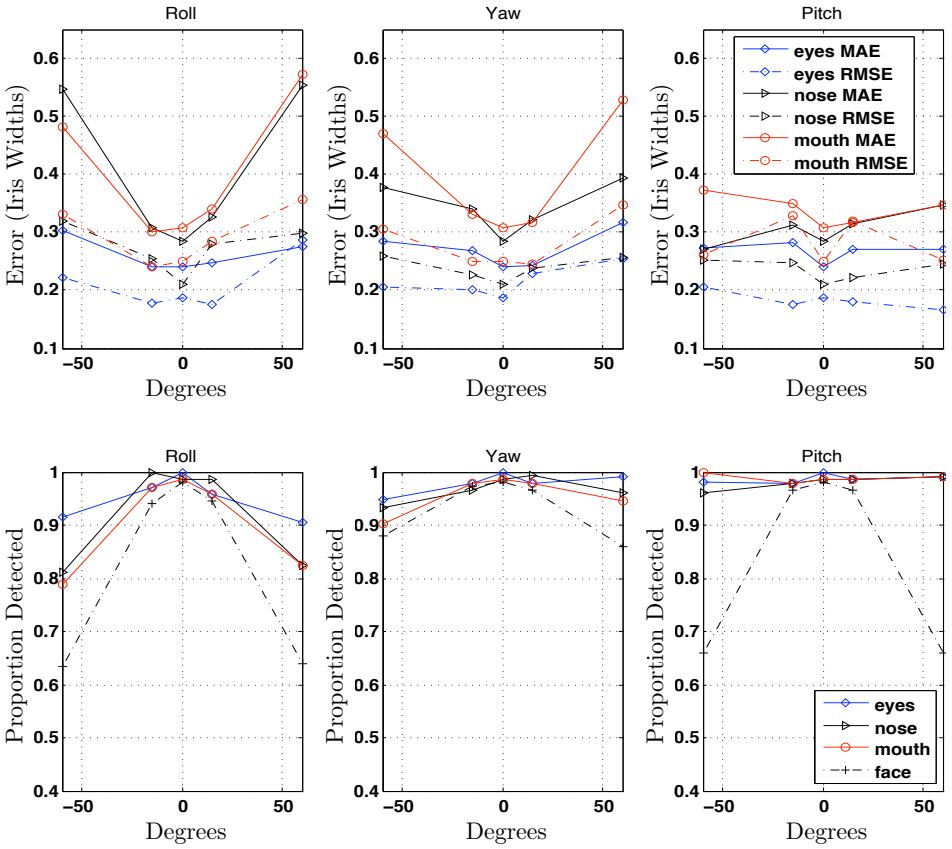


Fig. 13. Effects of Pose on Performance.

5. Prior Work

We compared the performance of our system with the best performing systems published to date [5, 3, 6], including a previous version of our system [7].

In all of the figures we used the following acronyms: (1) ADA-AAM represents the work of [5], Adaboost with Active Appearance Model. (2) SVM represents the SVM approarch in [3]. (3) "Bayesian", "Regression" and "Discriminative" represents the 3 different methods presented in [6] as: "Bayesian", "regression", and "discriminative". (4) CDI represents our approach, Context Dependent Inference. The system in [7] is named "CDI-0", and the current system "CDI-1."

5.1. *Adaboost with Active Appearance Model*

Cristinancce et al. [5] present results on an approach similar to the one presented here: A Viola-Jones style face detector is used to locate faces on the image plane.

Once a a face is localized, a similar style detector is applied to selected regions of the face. In addition, they apply an active appearance model (AAM) to the candidate points to refine the final location estimate, and infer missing points. Unfortunately we do not have access to the XM2VTS dataset, so we can only compare our results on the FERET database with their results on the XM2VTS dataset. These two dataset are considered comparable in difficulty. With this caveat in mind, Figure 14 indicates that our system is likely to be much more accurate than Cristinancce et al. [a].

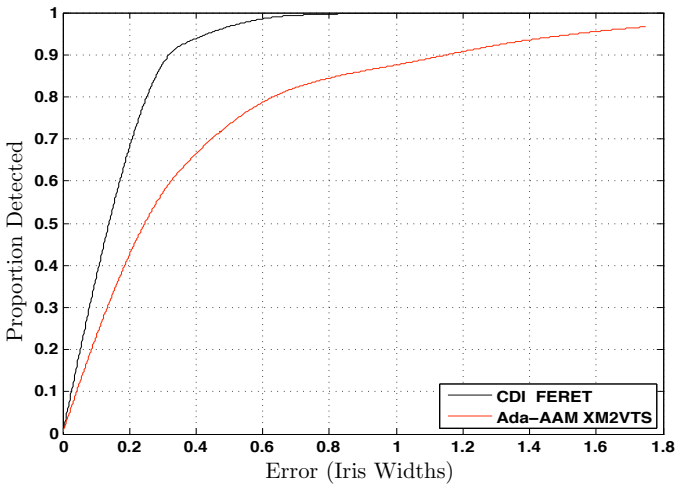


Fig. 14. Comparison between the Appearance Model of [5], indicated by Adaboost-AAM, and this current system, labeled CDI for Context Dependent Inference. Performance measured on the XM2VTS database by [5] and measured on the FERET database for this system.

5.2. *Multi-Module SVM*

P. Campadelli et al. present a general-to-specific model for eye detection that can be applied to the output of any face detector that returns a rough estimate of the face location [3]. Once the face has been detected, eye localization is performed in a two step process by Support Vector Machine (SVM) modules: 1) the eye detector and 2) the eye localizer. The first SVM module performs a rough estimation of eye location by evaluating a subset of points in the face region. The second SVM module is then applied to the candidate points from the first step to refine localization accuracy.

[a]Results shown for [5] were extracted from published graphs, thus the exact numerical values may differ slightly.

They report the system to be robust to background clutter, moderate illumination variation and head rotation up to 20^o both in and out of plane. Figure 15 shows that our system significantly outperformed the SVM system [3] on the FERET and BioID databases [b].

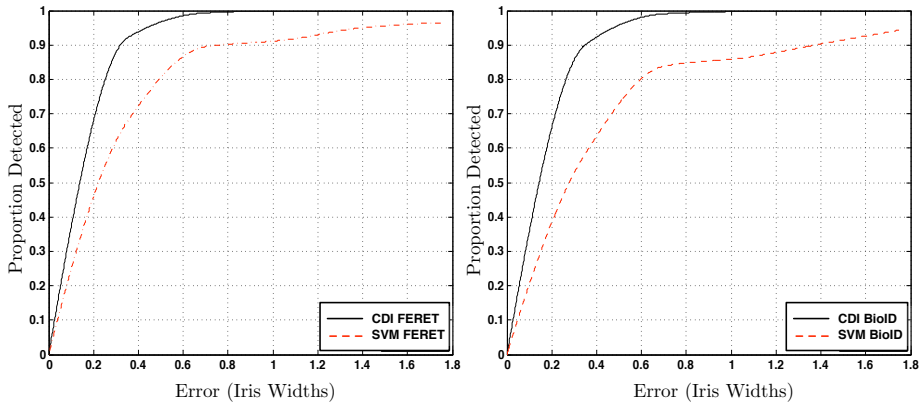


Fig. 15. Comparison with multi-module SVM of [3], indicated by SVM, and the current system, labeled CDI. Performance is given for FERET database (left) and BioID database (right).

5.3. *Bayesian, Regression and Discriminative Classifier*

Everingham et al. [6] compare three approaches (regression, Bayesian and discriminative classifier) to eye localization. Their results indicate that the Bayesian approach performs best, detecting 90% of the eyes within approximately 0.329 iris widths. This level of performance was described as "remarkable" in a recently published survey of the field [4]. Our system localized 92% of the eyes within 0.329 iris widths. While in absolute values a 2 % improvement may appear small, it is in fact a 20 % reduction of the state-of the-art error rate (see Figure 16). Figure 16 shows a comparisons of our systems with the results in [6] on the FERET database[c].

5.4. *Previous Version of Context Dependent Inference*

The current approach to feature detection was based on an eye detection system that we had previously developed in [7]. We compare performance of the current

[b]Results shown for [3] were extracted from published graphs, thus the exact numerical values may differ slightly.
[c]Results shown for [6] were extracted from published graphs, thus the exact numerical values may differ slightly.

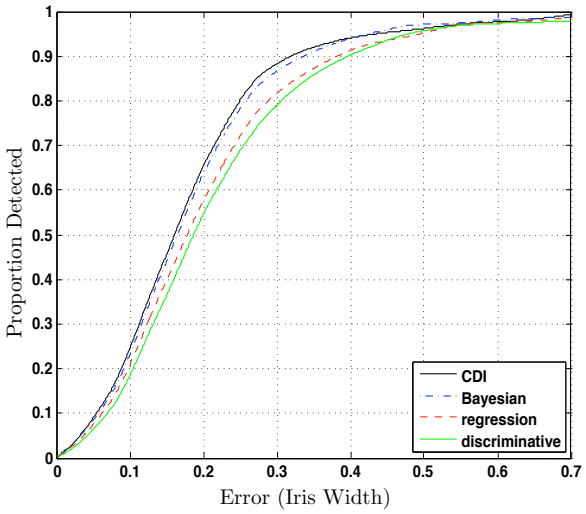


Fig. 16. Comparison with regression, discriminative classifier and Bayesian approach presented by [6], indicated by regression, discriminative and Bayesian figure labels, and this current system, labeled CDI for Context Dependent Inference. Performance given for the FERET database.

system to [7] on the FERET, BioID and GENKI databases. "CDI-0" indicates [7] and "CDI-1" indicates our current system. Figure 17 shows that the current system is a significant improvement over the previous work in eye detection. On FERET, we detect 94.0% of eyes within 0.4 iris widths, while the previous system only detected 74.8% at this level. On BioID we detect 92.8% within 0.4 iris widths while the previous system only detected 70.9% within this distance. On the GENKI database we detect 75.9% within 0.4 iris widths while the previous system manages only 56.2% within the same distance.

5.5. *Comparison with Human Performance*

In the previous sections we compared performance of our automated system using the human labels as ground truth. However, since the human labels may be noisy, it is unclear whether the system is already achieving the best possible performance given the quality of the labels. To this effect 7 people hand labeled the eyes, nose and mouth in 100 randomly selected images from the GENKI database. This test set was also labeled automatically by our feature detection system. The pre-existing hand labels were used as ground truth. Tables 2 and 3 display these results. Human labelers outperformed the automated system, by a factor of about 1.7 for the eyes, a factor of 2 for the nose, and a factor of 3 for the mouth. While this indicates that there is room for improvement, the performance of the eye detectors is already remarkably close to that of humans and is certainly useful for practical applications.

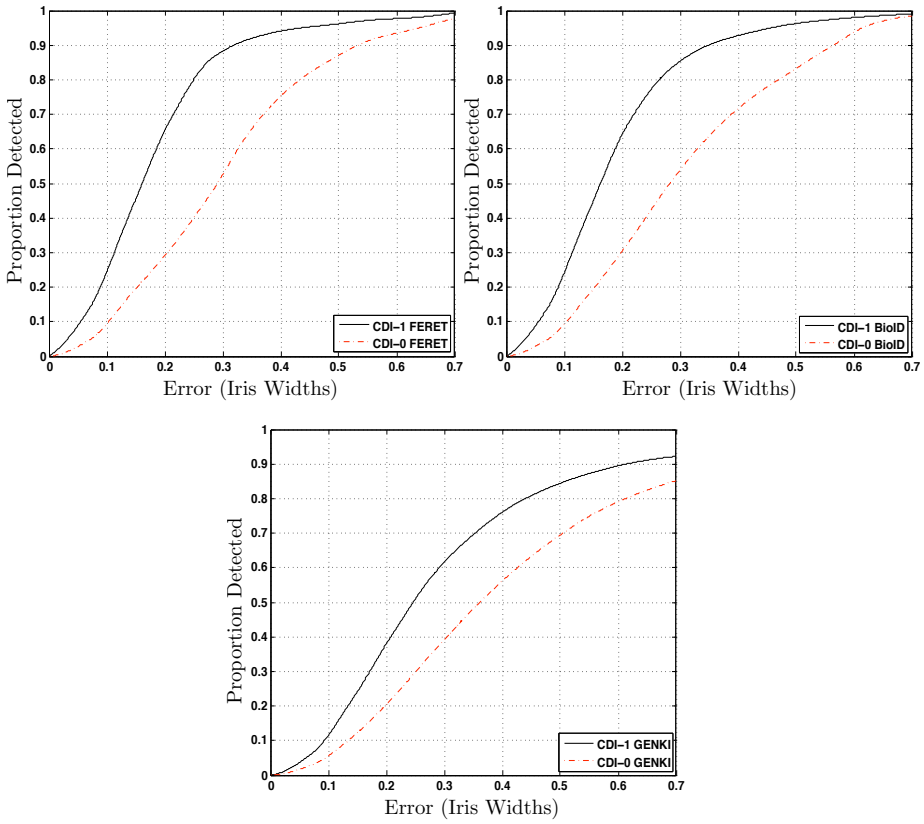


Fig. 17. Comparison with [7], indicated by CDI-0, with the current work, indicated by CDI-1. Performance given for FERET database (top left), the BioID database (top right) and the GENKI database (bottom center).

Table 2: Human vs. machine: MAE

Feature	Human	Machine	Machine/ Human
Left eye	0.1173	0.1938	1.65
Nose	0.1621	0.3152	1.95
Mouth	0.1266	0.3853	3.04

Table 3: Human vs. machine: RMSE

Feature	Human	Machine	Machine/ Human
Left eye	0.0811	0.1476	1.82
Nose	0.1070	0.3969	3.71
Mouth	0.0791	0.7502	9.48

6. Conclusions

Robust and accurate localization of facial features is critical for an emerging generation of practical applications of machine perception technology applied to the human face. A key challenge in feature detection architectures is the need to address

an inherent tradeoff between robustness and precision. Detectors that are robust to variations in illumination and imaging conditions do not perform localization well. Detectors trained to localize features precisely tend to produce a large number of false alarms.

Here we approach this problem by refining a context dependent inference architecture previously proposed in [7]. First robust detectors are used to detect the general context in which features appear and then precise detectors are used that operate within that context.

We presented empirical studies of the different factors affecting performance within this architecture. These experiments showed: (1) Significant performance levels can be achieved with 1000-5000 training examples. (2) The negative examples within the *ROI* should be highly dissimilar from the positive examples. (3) The optimal receptive fields are relatively large, from 7.7 to 9.8 iris widths. (4) Sampling at a higher rate than the resolution of the detector reduces error. (5) A high degree of accuracy can be achieved with only fifty to one-hundred box filters. (6) Careful integration of the output of detectors provides significant performance gains. (7) Eye detectors are relatively robust to pose variation, sustaining good performance levels with deviations from frontal pose of up to 60 degrees. (8) The performance levels of the eye detectors approximate human levels of accuracy, and are ready for practical applications.

The approach explored in this document signficantly outperformed previous state-of-the-art methods [6, 5, 3, 11] in terms of detection and localization accuracy. Our experience using a large database of images from the World Wide Web leads us to believe that the current databases used in the literature are too easy and not particularly useful to assess performance in the challenging situations needed for practical applications. Our work indicates that training with images representative of "real world" conditions is necessary to provide practical levels of performance. Collection and standardization of such databases may help accelerate progress in the field.

Acknowledgments

The study was funded by the UC Discovery Grant #10202 and by the NSF Science of Learning Center grant SBE-0542013.

References

1. The BANCA Database. `http://www.ee.surrey.ac.uk/CVSSP/banca/`.
2. The BioID Database. `http://www.bioid.com/downloads/facedb`.
3. P. Campadelli, R. Lanzarotti, and G. Lipori. Percise eye localization through a general-to-specific model definition. In *In. Proceedings of the British Machine Vision Conference*, 2006.
4. P. Campadelli, R. Lanzarotti, and G. Lipori. Eye localization: a survey. In

The fundamentals of verbal and non verbal communication and the biometrical issues, volume vol. 18 of *NATO Science Series*, Amsterdam 2007.

5. D. Cristinacce and T. Coots. Facial feature detection using adaboost with shape constraints. In *In Proceedings of the British Machine Vision Conference*, 2003.
6. M. Everingham and A. Zisserman. Regression and classification approaches to eye localization. In *Proceedings of the 7th International Conference of Automatic Face and Gesture*, 2006.
7. I. Fasel, B. Fortenberry, and J. Movellan. A generative framework for real time object detection and classification. In *Computer Vision and Image Understanding*, volume 98, pages 182–210, 2005.
8. I. R. Fasel, M. Stewart Bartlett, and J. R. Movellan. A comparison of Gabor filter methods for automatic detection of facial landmarks. In *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition*. Washington DC, 2002.
9. J Friedman, T Hastie, and R Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28(2):337–374, 2000.
10. J. Huang and H. Wechsler. Eye detection using optimal wavelet packets and radial basis functions (RFFs). *International Journal of Pattern Recognition and Artificial Intelligence*, 7(13), 1999.
11. O. Jesorsky, K.J. Kirchberg, and R.W. Frischholz. Robust face detection using the hausdorff distance. In *In Proceedings of Audio and Video based Person Authentication*, 2001.
12. R. Kothari and J. Mitchell. Detection of eye locations in unconstrained visual images. *ICIP96*, 1996.
13. T. Leung, M. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *5th International Conference on Computer Vision*, 1995.
14. Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. In *In Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 339–344, 2004.
15. P.J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithims. *Image and Vision Computing*, 16(5):295–306, 1998.
16. H. Rowely, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(20):23–28, 1998.
17. G. Shakhnarovich, P. Viola, and B. Moghaddam. A unified learning framework for real-time face detection and classification. *International Conference on Automatic Face and Gesture Recognition*, 2002.
18. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple feature. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
19. P. Wang, M. Green, Q. Ji, and J. Wayman. Automatic eye detection and its val-

idation. In *In. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

20. J. Whitehill, M Bartlett, G. Littlewort, I. Fasel, and J. Movellan. Developing a practical smile detector. (*under review*).
21. L. Wiskott, J. M. Fellous, N. Krüger, and C von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.
22. The XM2VTS Database. `http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb`.
23. A. L. Yuille and H. H. Bulthoff. Bayesian decision theory and psychophysics. Technical Report 2, Max Planck Institut fur Biologische Kybernetik, Arbeitsgruppe Bulthoff, 1993.
24. Z. H. Zhou and Y. Jiang. Projection functions for eye detection. *Pattern Recognition Journal*, 2004.