IMPROVED FACE FINDING IN VISUALLY CHALLENGING ENVIRONMENTS

*Jintao Jiang**, *Gerasimos Potamianos, Giridharan Iyengar*

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

Emails: jjt@icsl.ucla.edu , {gpotam,giyengar}@us.ibm.com

ABSTRACT

Finding faces in visually challenging environments is crucial to many applications, such as audio-visual automatic speech recognition, video indexing, person recognition, and video surveillance. In this study, we investigate several algorithms to improve face detection accuracy in visually challenging environments using the IBM appearance based face detection system. The algorithms considered are trainable skintone pre-screening, Hamming windowing of the face images, DCT coefficient selection, and the AdaBoost technique. When these methods are combined, an up to 68% relative reduction in face detection error is observed on visually challenging datasets.

1. INTRODUCTION

Robust face detection is the first and indispensable step for many applications, such as audio-visual automatic speech recognition (AVASR), video indexing, user interfaces, and video surveillance. Robust face detection is a difficult problem, especially in visually challenging (realistic and non-ideal) environments. Such cases are of particular interest to our work on AVASR, as we target the practical deployment of this technology [1, 2]. Visual speech has been shown to improve ASR in noise for "visually clean" data [1, 3], however obtaining such data is not always feasible [4]. Indeed, real applications often encounter visually challenging environments, where variations in the speaker's head pose and appearance, environment lighting and background, as well as in the video acquisition sensor's quality degrade the AVASR system accuracy. For example, the visual-only speech recognition error rate of connected digit strings increases significantly from studio-like environments (29.5%) to the more challenging office (46.1%) and moving-car (66.7%) domains (see [4] for details on the datasets and results). Such effects can be mostly attributed to poor face detection, thus motivating our work in this paper. Towards this end, we aim at improving face detection accuracy by investigating several algorithms to augment the current IBM face detection system. Clearly, we are particularly interested in the performance of these techniques in visually challenging domains.

There exist two main approaches for face detection [5, 6]: Geometric feature based matching and appearance based matching Most systems belong to the second category and can be further classified into color segmentation and statistical modeling based systems. The first rely on skin color modeling, using for example Gaussian-like distributions [7], or empirical skintone tables [8]. Examples of statistical modeling techniques include neural networks [9], dynamic link matching [10], Fisher's linear discriminant analysis (LDA) [1, 8], support vector machines [11], eigenfaces [12], hidden Markov models [13], embedded Bayesian networks [14], and Gaussian mixture model (GMM) classifiers [15]. The above use so-called "strong learners" for face detection. An alternative technique, first proposed by Viola and Jones [16], employs a quite different but very fast face detection algorithm based on AdaBoost [17] and feature selection ("weak learner").

The face detection approach in this paper is based on the existing IBM face detection system that has evolved from using LDA to employing GMMs for classification, as described in prior work [2]. There, we demonstrated large improvements when considering GMMs instead of LDA both in face detection and in the resulting AVASR accuracy. Such improvements were achieved without adding significant modeling and computational complexity overhead. In this paper, we incorporate several additional algorithms into the GMM based system, namely a Gaussian based trainable skintone model and the AdaBoost approach [17]. Furthermore, we introduce 2-D Hamming windowing applied on the candidate face images, and we propose a new selection algorithm of the discrete cosine transform (DCT) coefficients employed in our face detection system, that is a combination of the Bayesian information criterion [18] and a two-class correlation method. We then benchmark the proposed algorithms on three face detection tasks.

The paper is organized as follows: Section 2 describes the baseline face detection system [2], whereas Section 3 is devoted to the algorithms proposed to augment it. Section 4 describes the databases used in this study, with results presented in Section 5. Finally, a brief summary is given in Section 6.

2. THE BASELINE FACE DETECTION SYSTEM

Fig. 1 schematically depicts the baseline system, used for face detection in [2], and already being an improved version of the system in [1, 8]. Given an image, a pyramid is built to generate candidate face regions at different scales, so as to cover a variety of face sizes, since the face size is unknown in advance. Each such candidate generates a face vector that consists of the grey-level pixel values of its region, normalized to a rectangular template of size $M \times N$ pixels (typically 11×11), and in the case of the system in [2], projected to a lower dimension using a two-dimensional DCT. The resulting vector is scored by a two-class GMM classifier, or by an LDA in the original system of [8] – see the dotted lines in Fig. 1. If the score is high, a subsequent "distance from feature space" (DFFS) is applied to eliminate false faces [2, 8].

Training face samples are extracted from a limited number of video images that are manually annotated. To minimize mismatch between training and test data, the face selection box is slightly translated and rotated to produce additional samples (a total of ten

*Work performed while on internship at the IBM T.J. Watson Center. Jintao Jiang is currently with the House Ear Institute, Los Angeles, CA 90057, USA. The authors would like to acknowledge contributions to this work by Andrew W. Senior, Harriet Nock and Chalapathy Neti.

0-7803-9332-5/05/$20.00 ©2005 IEEE

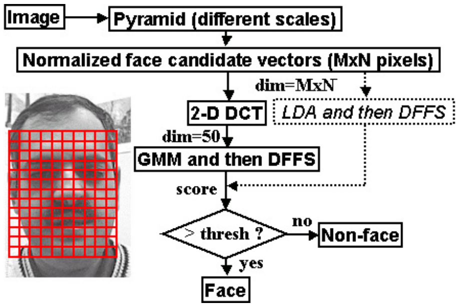

Fig. 1. The baseline IBM face detection system [2].

variations for each annotated face). The non-face samples are randomly chosen from the video images (but located away from the annotated faces), and ten random non-faces are generated for each image. For the GMM method, up to 50 mixture components are trained on 50-dimensional DCT face vectors.

3. FACE DETECTION ALGORITHMS

This section describes the algorithms considered in the paper for improving face detection robustness. In sequence, they are a trainable skintone model, Hamming windowing of the face candidate region, DCT coefficient selection of the face vectors, and the AdaBoost technique. All are considered on top of the baseline system of Section 2.

3.1. A trainable skintone model

The objective of skintone pre-screening is to limit the face search space and thus improve speed and accuracy. Although skintone varies across people, in general human skin color is different from most other objects. In [8], a simple color based thresholding scheme was used to detect skintone. The following range was first defined as the skintone region: Hue (35 to 240), chromaticity (6 to 240) and intensity (60 to 766). These values were mapped to a lookup table in the RGB space using specific rules [8]. As a result, for any given (G,B) pair, $R^{G,B}_{min}$ and $R^{G,B}_{max}$ were derived to define a skintone region as depicted in Fig. 2(a). This skintone table is not trained from a specific dataset and is preset for all camera conditions. As a result, it does not work well when there is image color distortion due to varying camera transfer functions. Of course, the table could be trained from data, however the model by default will still include non-skintone RGB points, in order to guarantee that all real skintone RGB points are inside the "polygon" region.

In this study, we follow a different approach to skintone modeling, motivated by work in [7]. There, it is demonstrated that the skintone distribution in the RGB space is Gaussian-like. We therefore consider a trainable 3-D Gaussian skintone model. Given a training set of skintone pixels, a 3×3 RGB covariance matrix $\sum_{rgb}$ and a 1×3 RGB mean vector $\mathbf{M}_{rgb}$ are obtained. Note that the effect of outliers (non-skintone pixels) is minimized automatically by the statistical processing. An additional advantage of the model is that it allows for easy adaptation by statistical techniques.

During testing, any given pixel is considered as skintone-like, if its RGB vector $\mathbf{P}_{rgb}$ satisfies

$$[\,\mathbf{P}_{rgb} - \mathbf{M}_{rgb}\,]\left(\sum\nolimits_{rgb}\right)^{-1}[\,\mathbf{P}_{rgb} - \mathbf{M}_{rgb}\,]^{\mathrm{T}} \le 6.0\;. \qquad (1)$$

The RGB space region defined by (1) is depicted in Fig. 2(b), and it is notably different from the table based distribution of Fig. 2(a). Face candidate regions containing less than 30% of skintone-like pixels (i.e. pixels that satisfy (1)) are classified as non-faces.

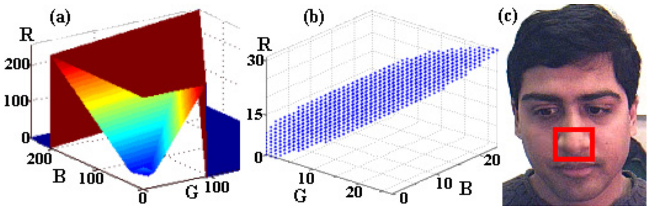

Fig. 2. Skintone distribution in the RGB space as (a) a table and (b) a Gaussian distribution. Data selection for Gaussian training of the skintone is depicted in (c).

Fig. 2(c) depicts typical input data used for training the Gaussian skintone model. The data selection box is centered at the nose with sides approximately 80% of the eye separation. Pixels inside that box are treated as skintone-like during the training phase.

3.2. Hamming windowing

The Hamming windowing technique helps smooth the boundaries of face candidate images that are obtained from a larger image. The face area covers eyes, nose, mouth, eyebrows, etc., and thus sometimes has sharp intensity changes. When face candidates are "cut" from an image, the boundaries usually are not smoothed, resulting in an un-smoothed 2-D DCT spectrum. Such spectrum could contain information about face identity or other conditions that are not of interest to person-independent face detection. Therefore, and similarly to ASR, a Hamming windowing technique is applied to the 2-D face candidate pixel intensities $\mathbf{I}_{i,j}$, as

$$\mathbf{I}_{i,j} \leftarrow \mathbf{I}_{i,j}\mathbf{H}^{M}_{i}\mathbf{H}^{N}_{j}\,,\ \text{where}\ \mathbf{H}^{K}_{k} = .54 + .46\cos\left(\pi\frac{k - \frac{K-1}{2}}{K+2}\right)$$

for $k = i, j$ and $K = M, N$.

3.3. DCT coefficient selection

In our face detection system, a 2-D DCT is applied on each face candidate, in order to produce a compressed image representation. In the baseline system [2], the resulting matrix of DCT coefficients is organized into a vector using a zig-zag scan, with only its first 50 coefficients used in classification (the energy term is excluded). This method guarantees that the low-frequency terms are selected, while ignoring the high-frequency components. However, such arbitrary selection of DCT coefficients may not be optimal, since, for example, human faces have special structure and thus result in specific DCT component distribution patterns. Here, we propose a combination of the Bayesian information criterion (BIC) [18] and a two-class correlation method for DCT coefficient selection.

3.3.1. BIC based DCT coefficient selection

The BIC is a well known statistical measure of separability between two classes, often employed for acoustic change detection [18], for example. Since face detection is a two-class problem, the BIC can be used to measure how well the face and non-face classes are separated. Here, the two classes are denoted by F=f, $\overline{f}$, respectively, each having N_F samples. The BIC is then

$$\text{BIC} = N\log\left|\sum\right| - N_f\log\left|\sum\nolimits_f\right| - N_{\overline{f}}\log\left|\sum\nolimits_{\overline{f}}\right|\;, \qquad (2)$$

where $N = N_f + N_{\overline{f}}$, and $\sum$, $\sum_F$ denote the covariances of all, or only the class-specific data vectors. Clearly, the larger the BIC value, the more separated the two classes are. However, the BIC value also depends on the spread magnitude of the two classes, as depicted in Fig. 3. There, the left case would result to a larger BIC value than the right one, even though the two classes are better separated in the right case.

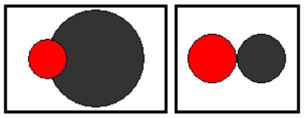

Fig. 3. Two classification cases.

3.3.2. *Two-class correlation method*

Another separability measure is the two-class correlation, which depends less on the magnitude of the two classes than the BIC. The method is similar to an LDA projection. As shown in (3), face and non-face vectors are placed together into a matrix and subsequently projected to one-dimensional vectors, ideally consisting of +1's for the face samples, and -1's for non-faces, namely:

$$\left[\mathbf{Dct}^{\overline{f}}_{1:N_{\overline{f}},1:MN}\;\mathbf{Dct}^{f}_{1:N_f,1:MN}\right]\mathbf{P}_{1:MN,1} \rightarrow \left[-\mathbf{1}^{\overline{f}}_{1:N_{\overline{f}},1}\,\mathbf{1}^{f}_{1:N_f,1}\right]. \quad (3)$$

Multilinear regression is used to define the projection vector $\mathbf{P}$. Of course, such a projection may not necessarily be successful. Therefore, we derive the correlation coefficient r between the actual projection (the left side of (3)) and the expected projection (the right side of (3)). In Fig. 3, the right case would have a larger correlation coefficient than the left one. The disadvantage of this method is that we may end up choosing many small-spread-magnitude components, vulnerable to noise.

3.3.3. *Combination of BIC and two-class correlation*

As stated in Sections 3.3.1 and 3.3.2, using only BIC values may result in inseparable DCT components selected, whereas using only two-class correlation may help choose small-spread-magnitude DCT coefficients. As a compromise, in this work, we propose to combine the two methods. For each DCT component i = 1,..., MN, we define its significance value S_i as

$$S_i = (\,\mathrm{BIC}_{all} - \mathrm{BIC}_{all\backslash i}\,) \cdot (\,r_{all} - r_{all\backslash i}\,)\,, \quad (4)$$

where $\mathrm{BIC}_{all\backslash i}$ and $r_{all\backslash i}$ are computed as in (2) and (3), respectively, but using $(MN-1)$-dimensional data vectors by excluding DCT component i. Assuming independence between the DCT coefficients, (4) is further simplified as

$$\left.\begin{array}{l}\mathrm{BIC}_{all} - \mathrm{BIC}_{all\backslash i} \approx \mathrm{BIC}_i \\ r_{all} - r_{all\backslash i} \approx r_i\end{array}\right\} \rightarrow S_i = \mathrm{BIC}_i \cdot r_i\;. \quad (5)$$

Based on (5), all MN values of S_i are computed and the top 50 are selected as the face vector elements fed into the GMM classifier.

3.4. AdaBoost

AdaBoost is a general algorithm that attempts to improve the performance of any classifier with more than 50% initial accuracy [17]. Under the AdaBoost framework, a series of learners are trained incrementally, each trying to handle "difficult" to classify samples by the previous learner, by assigning more weight to them. At the end, all learners are combined by weighting, in order to provide a stronger learner. The method is summarized below:

```
Given: $(x_1, y_1), \cdots, (x_m, y_m); x_i \in X, y_i \in \{-1, 1\}$
Initialize with equal weighting $D_1(i) = 1/m$.
For $t = 1, \cdots, T$ :
    • Train weak learner using distribution $D_t$.
    • Get weak hypothesis $h_t : X \rightarrow \Re$.
    • Choose $\alpha_t \in \Re$.                                  (6)
    • Update: $D_{t+1}(i) = D_t(i) \exp[-\alpha_t y_i h_t(x_i)] / Z_t$
      where $Z_t$ is a normalization factor (chosen
      so that $D_{t+1}$ will be a distribution).
Final hypothesis: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
```

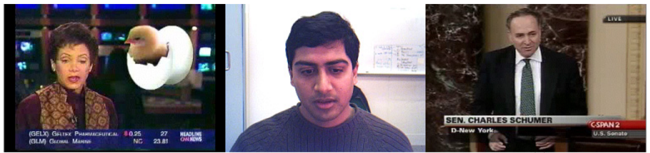

Fig. 4. Images from databases (a) CNN, (b) OFFICE, and (c) TREC.

In this work, five learners are trained for AdaBoost (T=5). A minor modification is made in the calculation of the weighting parameter α_t of (6), which is now given by

$$\alpha_t = \mathbf{.5}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \rightarrow \alpha_t = \mathbf{.25}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)\,. \quad (7)$$

In (7), ε_t denotes the training error of each weak learner, with the distribution weighting considered. Such a modification is determined empirically, and it produces a less aggressive distribution updating. This may be due to the fact that the initial weak learner is already very good (typically $\varepsilon_1 < 0.15$).

4. DATABASES

Three databases representing different visually challenging environments are used in this paper to benchmark the proposed face detection algorithms. The first corpus consists of video recordings of CNN programs, and will therefore be referred to as the "CNN" set. Such data have frequent scene changes and non-frontal or tilting faces. The second database has been recorded using a laptop based audio-visual data collection prototype with a USB 2.0 inexpensive web-cam, originally for AVASR experiments [1, 4]. The 109 database subjects were recorded in their own offices with varying lighting, background, and head-pose. This set will be referred to as "OFFICE". The third database contains mostly CSPAN broadcast news programs with some ABC and CNN segments, and it is part of the "TREC" corpus. Thus, this set also includes non-anchor speakers with tilting, small, or non-frontal faces. Representative frames from all sets are depicted in Fig. 4.

For each database, two sets of face images are manually annotated for training and testing. Their details are depicted in Table 1, together with information on the frame and average face sizes, the latter expressed by the eye separation. Notice that the TREC database is the most challenging in terms of face size. Concerning head pose (not quantified in the table), TREC is again the most challenging set, followed by CNN. On the other hand, face images in the OFFICE corpus are rather upright, however they have the largest color distortion due to the web-cam used.

5. RESULTS

We now proceed to benchmark the performance of the discussed face detection algorithms on the test sets of the three visually challenging databases of Section 4. A number of experimental results are depicted in Table 2, given in terms of face detection accuracy. A face is considered detected if the location error is within 20% of the annotated eye separation. In case of multiple detected faces in a frame, only the one with the highest statistical score is considered.

Table 1. Comparison of the three databases (E: eye separation in pixels). Numbers in parenthesis are for training and testing, respectively.

Database	Image size	Faces	mean(E)	min(E)
CNN	704×480	(1227, 303)	(70,70)	(28,40)
OFFICE	320×240	(1368, 253)	(50,50)	(30,33)
TREC	352×240	(1345, 292)	(26,29)	(20,25)

Table 2. Face detection accuracy (%) on the three database test sets using the algorithms discussed in this paper: ST: skintone table; TS: trainable skintone; HW: Hamming windowing; BST: AdaBoost; DS: DCT coefficient selection. Two slightly different face template sizes are considered, depicted inside parentheses.

Algorithm Dataset →	CNN	OFFICE	TREC
LDA (11x11)	75	91	46
GMM (11x11)	83	**97**	78
GMM+ST (11x11) – Baseline	**89**	71	**81**
GMM+TS (11x11)	90	98	82
GMM+TS (14x11)	89	99	82
GMM+TS+HW (14x11)	92	97	87
GMM+TS+HW+DS (14x11)	92	96	96
GMM+TS+HW+BST (14x11)	94	100	91
GMM+TS+HW+DS+BST (14x11)	**92**	**99**	**94**

We first compare the LDA and GMM based approaches. In line with experiments reported in [2], GMMs dramatically outperform LDA consistently across all sets. For example, on TREC, the accuracy improves from 46% to 78%, an approximately 60% relative reduction in error. Results further improve on the CNN and TREC sets using the skintone lookup table, however the method is not appropriate for the OFFICE set, because of color distortion in the data. We consider the best of the two results (with or without the table lookup) as the face detection accuracy of the "baseline" system.

Following the presentation sequence of the algorithms discussed in this paper for face detection, we first consider the trainable skintone model. We notice that although it barely improves performance over the best "baseline" accuracy, it is robust across all sets. Next, we slightly adjust the template face size, from 11×11 to 14×11 pixels. This follows our work in [2], where the rectangular template was deemed beneficial to AVASR. Although there is no significant benefit to the face detection accuracy in the three sets, in the remaining of the results we consider only the larger template.

Adding Hamming windowing to the system improves performance significantly for the TREC and CNN sets, with some degradation in the OFFICE case. Furthermore, DCT coefficient selection (after Hamming windowing) works extremely well for the TREC database that is the most visually challenging, with no major difference observed in the other two sets. AdaBoost, on the other hand, consistently improves on Hamming windowing for all three databases. Finally, combining all the algorithms described in Section 3 produces robust face detection on all three databases, with significant improvement over the baseline. In particular for the TREC set, we obtain a 94% accuracy, which corresponds to a 68% relative reduction in error, as compared to the 81% accuracy of the baseline.

6. SUMMARY

This work focuses on robust face detection in visually challenging environments. Reported results indicate that the trainable skintone model and Hamming windowing are effective and simple to implement. The DCT component selection works well with the most challenging TREC set, suggesting that zig-zag coefficient selection is indeed not optimal and that various faces have different patterns of important DCT components. The results also show that AdaBoost, when used with Hamming windowing, results in further improvements. Combining all the discussed algorithms produces a face detection system with satisfactory accuracy and significantly better performance than our baseline.

It is interesting to note that when AdaBoost is combined with the DCT coefficient selection algorithm, the system can function as a feature selection procedure similar to the one in [16]. This way variations in the face samples can be better accounted for, with each weak learner possibly using specific patterns of DCT components, and thus having a specific representation of training face images.

The proposed system can be further evaluated by comparing it to other popular face detectors such as the ones in [9, 16], and by testing on a standard set such as the combined MIT/CMU database. It would also be interesting to investigate whether the reported improvements translate to better AVASR, especially in the broadcast news domain.

7. REFERENCES

[1] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior, "Recent advances in the automatic recognition of audio-visual speech," *Proc. IEEE*, 91: 1306–1326, 2003.

[2] J. Jiang, G. Potamianos, H. Nock, G. Iyengar, and C. Neti, "Improved face and feature finding for audio-visual speech recognition in visually challenging environments," in *Proc. ICASSP*, vol. 5, pp. 873–876, 2004.

[3] E.D. Petajan, *Automatic Lipreading to Enhance Speech Recognition*, Ph.D. thesis, University of Illinois, Urbana-Champaign, IL, 1984.

[4] G. Potamianos and C. Neti, "Audio-visual speech recognition in challenging environments," in *Proc. Eurospeech*, pp. 1293–1296, 2003.

[5] M.H. Yang, D.J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 24: 34–58, 2002.

[6] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Patt. Anal. Mach. Intell.*, 15: 1042–1052, 1993.

[7] J. Yang, W. Lu, and A. Waibel, "Skin-color modeling and adaptation," in *Proc. Asian Conf. Comp. Vision*, vol. 2, pp. 687–694, 1997.

[8] A.W. Senior, "Face and feature finding for a face recognition system," in *Proc. Int. Conf. Audio-Video based Biometric Person Authent.*, pp. 154–159, 1999.

[9] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Patt. Anal. Mach. Intell.*, 20: 23–38, 1998.

[10] L. Wiskott and C. von der Malsburg, "Recognizing faces by dynamic link matching," in *Proc. ICANN*, pp. 347–352, 1995.

[11] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. CVPR*, 1997.

[12] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Patt. Anal. Mach. Intell.*, 19: 711–720, 1997.

[13] A.V. Nefian and M.H. Hayes, "Face detection and recognition using hidden Markov models," in *Proc. ICIP*, 1998.

[14] A.V. Nefian, "Embedded Bayesian networks for face recognition," in *Proc. ICME*, 2002.

[15] K. Sung and T. Poggio, "Example-based learning for view-based face detection," *IEEE Trans. Patt. Anal. Mach. Intell.*, 20: 39–51, 1998.

[16] P. Viola and M. Jones, "Robust real-time object detection," in *Proc. Int. Workshop Stat. and Comp. Theory of Vision-Modeling, Learning, Computing, and Sampling*, 2001.

[17] R.E. Schapire, "Theoretical views of boosting," in *Proc. European Conf. Comp. Learning Theory*, pp. 1–10, 1999.

[18] S.S. Chen, E.M. Eide, M.J.F. Gales, R.A. Gopinath, D. Kanevsky, and P. Olsen, "Automatic transcription of broadcast news," *Speech Comm.*, 37: 69–87, 2002.