Pre-print of article that will appear at WACV 2012.

For Your Eyes Only

Brian Heflin* Walter Scheirer T.E. Boult *
bheflin@securics.com wscheirer@securics.com tboult@vast.uccs.edu
Securics Inc and University of Colorado Colorado Springs

Abstract

In this paper, we take a look at an enhanced approach for eye detection under difficult acquisition circumstances such as low-light, distance, pose variation, and blur. We present a novel correlation filter based eye detection pipeline that is specifically designed to reduce face alignment errors, thereby increasing eye localization accuracy and ultimately face recognition accuracy. The accuracy of our eye detector is validated using data derived from the Labeled Faces in the Wild (LFW) and the Face Detection on Hard Datasets Competition 2011 (FDHD) sets. The results on the LFW dataset also show that the proposed algorithm exhibits enhanced performance, compared to another correlation filter based detector, and that a considerable increase in face recognition accuracy may be achieved by focusing more effort on the eye localization stage of the face recognition process. Our results on the FDHD dataset show that our eye detector exhibits superior performance, compared to 11 different state-of-the-art algorithms, on the entire set of difficult data without any per set modifications to our detection or preprocessing algorithms. The immediate application of eye detection is automatic face recognition, though many good applications exist in other areas, including medical research, training simulators, communication systems for the disabled, and automotive engineering.

1. Introduction

Facial recognition technology is currently one of the fastest growing applications among the biometric technologies accepted worldwide [16]. In addition, it is the only biometric modality that can be used reliably at long ranges, and in a covert manner for surveillance applications since a subject's face can be easily captured by a video camera. *Eye detection* is a necessary processing step for many face recognition algorithms. For some of these algorithms, the eyes serve as reference points to locate other significant features on the face, such as the nose and mouth. Other face recognition algorithms rely substantially on accurate facial alignment before recognition, which is usually conducted using the subject's left and right eye positions. Facial alignment is an image registration problem, where the face needs to be deformed to match against other face images in the database. Inaccurate alignment is a well-known and persistent complication, and previous studies [15, 25] have shown that it has a large impact on recognition accuracy. Riopka and Boult showed [15] that even with ideal data, eye localization has a significant quantitative impact on the accuracy of several face recognition algorithms. In the classic study of Phillips et al. [12], several face recognition algorithms with and without manual eye alignment were also evaluated. The experiments of Phillips et al. also showed that algorithms which use manual facial alignment always outperformed algorithms with automatic facial alignment.

In this paper we present our enhanced eye detector algorithm, which builds on the work of our previous system [20]. Specifically, we present a novel algorithm that is designed to reduce face alignment errors, thereby increasing eye localization accuracy and ultimately face recognition accuracy. The method we propose employs an "average face" Unconstrained Minimum Average Correlation Energy (UMACE) filter and facial symmetry test to verify whether the face has been properly aligned using the detected eye coordinates.

Until recently, many researchers used controlled datasets (with fixed levels of lighting and blur) to develop and evaluate their eye detectors. While these datasets are useful in the creation and testing of an eye detector, they give little indication of how a detector will perform in difficult or uncontrolled circumstances. Performing eye detection in unconstrained environments has not been entirely solved because of the fundamental difficulties of various factors in the real world such as resolution, pose (in-plane and out-of-plane rotation), scale, illumination changes, occlusion, and atmospheric & motion blur. Our results on the LFW [7] and FDHD [8] datasets show that while our eye detector exhibits excellent performance on the entire set of difficult data, many of the other commercial and non-commercial eye detection algorithms performed poorly on the uncon-

*This work was supported by ONR STTR N00014-07-M-0421 and ONR MURI N00014-08-1-0638

strained real-world data.

We organize the rest of the paper as follows. In Sec. 2, we take a brief survey of the existing literature related to automatic eye detection. In Sec. 3, we introduce our novel correlation filter approach for feature detection, including a new technique to decrease facial alignment errors. Our experimental protocol is defined in Sec. 4, followed by a series of experiments to evaluate our eye detection approach using a subset of the LFW dataset and the FDHD dataset. The paper concludes in Sec. 6.

2. Related Work

Feature-based learning methods are prevalent in the literature, and demonstrate reasonable performance. Leite et al. [10] consider PCA features derived from the eyes, which are used as inputs to a neural network learning system. Using a dataset of 240 images of 40 different full frontal faces, this technique is shown to be as accurate as several other popular eye detection algorithms. Shuo and Liu [22] use color information and wavelet features together with a new efficient Support Vector Machine (eSVM) to locate eyes. The approach consists of two steps: selection of possible eye candidate regions using a color distribution analysis in YcbCr color space, followed by the application of 2D Haar wavelets to the image for multi-scale image representations and then PCA for dimensionality reduction, with eSVM detection. Wang et al. [25] use normalized eye images projected onto weighted eigenspace terrain features as features for an SVM learning system. Jin et al. [9] use a recursive non-parametric discriminant feature as input to an AdaBoost learning system. Eye characteristics such as edge and color distributions [6, 23] have also been used as features to identify the eye area.

The features these learning methods use are only efficient in high contrast images and lack the accuracy needed for real world applications such as surveillance. Correlation filters are a nice alternative approach, having the advantage of being able to capture general structure under many different circumstances, while suppressing false detections and missed detections. Brunelli and Poggio [4] first demonstrated the feasibility of correlation filters for the specific problem of eye detection, while Bolme et al. [2] introduced a more sophisticated class of filters that are more insensitive to over-fitting during training, more flexible towards training data selection, and more robust to structured backgrounds. In our previous work [20], we presented an adaptive average correlation energy (AACE) filter that has the benefits of prior correlation approaches and incorporates a blur model directly into the filter. We discuss our correlation filter designed for eye detection in detail in Sec. 3.

3. The Eye Detection Pipeline

3.1. Lighting Normalization

Lighting normalization is a vital first step in our eye detection pipeline. For our lighting normalization algorithm, We use a modified version of the Self-Quotient Illumination (SQI) lighting normalization algorithm, based on the work of Chen et al. [5]. The SQI image is formed by dividing the original face image $f(x, y)$ with the original image convolved with a Gaussian function $G(x, y)$ that acts as a smoothing kernel function $S(x, y)$:

$$Q(x,y) = \frac{f(x,y)}{S(x,y)} = \frac{f(x,y)}{G(x,y) \otimes f(x,y)} \quad (1)$$

However, the SQI algorithm presented by Chen et al. [5] does not always work effectively in all situations. For example, if the smoothed image $S(x, y)$ is very close to the original image $f(x, y)$ the ratio of most pixels are very close to 1. Therefore, features of the original image cannot be extracted efficiently. To compensate for this effect, we use an isotropic filter [24] to produce the smoothed image. The isotropic filter increases the ratio of the feature pixels to be greater or less than 1, thereby allowing more features to be extracted from the image. Additionally, we perform a contrast stretch before the filtering to force a larger separation of bright and dark areas in the image.

(a) (b) (c) (d) (e)

Figure 1. An example of SQI lighting normalization. (a) Original image (b) Gamma corrected & contrast stretched image (c) Smoothed image (d) Quotient image (e) Normalized quotient image.

The subsequent task of the lighting normalization method is to normalize $Q(x, y)$ to have pixel intensity between 0 and 1, and to again increase the contrast of the image by applying a linear transformation function

$$Q'(x,y) = \frac{Q(x,y) - Q_{min}}{Q_{max} - Q_{min}} \quad (2)$$

$$Q_{norm}(x,y) = 1 - e^{-\frac{Q'(x,y)}{E(Q'(x,y))}} \quad (3)$$

where Q_{max} and Q_{min} are maximum and minimum values of Q respectively, and $E(.)$ is a mean value. Therefore, Q_{norm} is a normalized Gaussian quotient image (an example is shown in Fig. 1(e)) used for eye detection.

3.2. Eye Region Segmentation

A common method to accurately find eyes is to first detect the face region and then search for the eyes in the subregion of the detected face. Many eye detection algorithms assume that the geometry of the cropped facial image is frontal and therefore their associated cropping algorithm uses fixed coordinates to extract the potential eye region. While this method is effective, face detectors such as the one built into the OpenCV library [3] can return faces that are not purely frontal which can lead to erroneous eye region cropping. For us to facilitate eye detection beyond frontal images it then becomes necessary to have a cropping algorithm that can weakly detect the eyes and return that potential eye region for verification using our correlation based eye detector. Our cropping algorithm begins with skin detection.

In grayscale images skin and non-skin areas are only distinguishable by their luminosity and/or texture. However, their respective intensity distributions show significant overlap thereby making per pixel decisions to classify a pixel as skin or non-skin very error prone. The skin detection algorithm presented below, based on the work of Pierrard and Vetter [13], operates on local neighborhoods by repeatedly matching the neighborhoods around the unprocessed pixels against a sample texture from the image. We define I^{tgt} as the target image for which the similarity should be computed. We further denote with I^{src} a source image and I^{seed} an associated mask defining the texture sample region. The similarity can then be computed for each pixel $p \in I^{tgt}$ independently by taking its local neighborhood N_p^{tgt} and searching within the seed region of I^{src} for the best matching pair $q \in I^{src}$. The texture similarity error per target pixel is defined by Eq. 4:

$$E_{ts}(p) := \min_{q|N_q^{src} \subset (I^{src} \cap I^{seed})} ||N_p^{tgt} - N_q^{src}||_{SSD} \quad (4)$$

However, this equation does not take the statistics of the sample texture into account. In order to determine how likely a target pixel may originate from this texture we are computing the k-nearest-neighbors to $p \in I^{tgt}$. The error $ts \in E^k$, analogous to Eq. 4, is defined as the average of the corresponding closest-patch distances. This extraction method has been adapted to our specific segmentation problem. We have chosen the upper cheek area as the facial area that is unlikely to contain outliers such as beards, glasses, etc. Since we are first applying the Viola-Jones face detector from the OpenCV library to the image, we are able to roughly determine the cheek area based on the returned face rectangle even if the face is not purely frontal. Once the cheek area is located, we segment an $N \times N$ patch, providing I^{seed}. The skin detection algorithm is then applied to neighborhoods in the cheek area under the assumption that the selected seed contains only skin. The output of $ts \in E^k$ inside of this area defines the range of matching errors that we can expect for similarly textured regions. The maximum of this range is used as a threshold for the rest of the image. Therefore, a pixel can be classified as skin or non-skin according to Eq. 5:

$$I^{skin}(p) := E_{ts}^k(p) \leq \max_{q \in I^{seed}} E_{ts}^k(q) \; ? \; 1 : \; 0 \quad (5)$$

Once the skin regions have been eliminated, the non-skin regions are grouped using a connected components algorithm [21]. Since we are scaling the image to a pre-defined size we can eliminate noise and large areas such as the edge of the face and hair using the area of the connected component region. We then search for a non-skin region around the expected area of the eye that is the closest to the expected region of the eye. Since we are scaling the facial image to a known size, we can use the size of the connected components region to assist in the selection of the region used for segmentation. Once the candidate eye region is located we segment a 64x64 pixel area, using the centroid of the selected region as the middle of the area to be cropped, for detection using our correlation eye filter. Sample segmentation results are shown in Fig. 2.

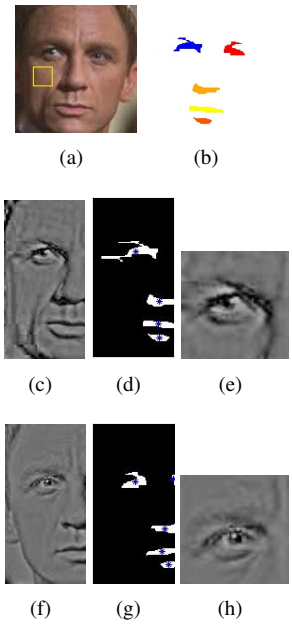


Figure 2. (a) Detected facial image with I^{seed} shown in the yellow box (b) Skin detection output showing non-skin regions (c) Left half of facial image (d) Connected components image of (c) showing centroid of each region (e) Segmented left eye region (f) Right half of facial image (g) Connected components image of (f) showing centroid of each region (h) Segmented right eye region.

3.3. UMACE Filter for Feature Detection

The correlation based eye detector is based on the Unconstrained Minimum Average Correlation Energy

(UMACE) filter [18]. Synthesis of the UMACE filter begins with cropping out regions of size 64x64 from the training data with the eye centered at coordinates (32,32). Our UMACE filter was synthesized with 3,000 such eye images form the FERET dataset [12]. A primary advantage of the UMACE filter over other types of correlation filters such as the Minimum Average Correlation Energy (MACE) filter [11] is that over-fitting of the training data is avoided by averaging the training images. After the eyes are cropped each eye region is transformed to its frequency domain representation using a 2D Fourier transform. Next, the average training images and the average of the power spectrum is calculated. The UMACE filter is synthesized using Eq. 6

$$h = D^{-1}m \tag{6}$$

where D is a diagonal matrix with the average power spectrum of the N training images placed along diagonal elements, and m is a column vector containing means of the Fourier transforms of the training images. UMACE filters are also attractive because they require less computation as compared to a MACE filter since inversion of only a diagonal matrix is needed. Furthermore, environmental degradations such as noise, motion, and atmospheric blur can convolved into the filter at run time on a image by image basis as in the AACE variant [20] using only a pointwise multiplication of the blur Optical Transfer Function (OTF) and the UMACE filter. One filter is designed for both the left and right eye since we flip the face image along the vertical axis once the left eye is found. The UMACE filter is stored in its frequency domain representation to eliminate another 2D Fourier transform before the correlation operation is performed. Since we are performing the correlation operation in the frequency domain the UMACE filter has to be preprocessed by a Hamming window to help reduce the edge effects in the spectrum. Our experiments indicated that windowing both the filter and input image decreased the accuracy of the UMACE eye detector. Furthermore, we found that training data taken under ideal conditions performed well for difficult detection scenarios such as low light and long distance acquisition when combined with the SQI lighting normalization described in Sec. 3.1.

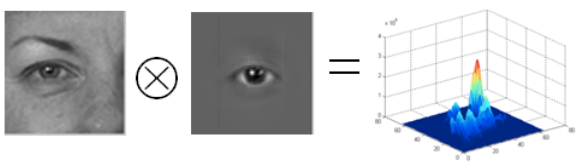


Figure 3. Example correlation output with the detected eye centered at coordinates (40,54)

Finally, to find the location of the eye, a 2D correlation operation is performed between the UMACE filter and the cropped face image. The global maximum or peak location is chosen as the detected eye location in the original image with the appropriate offsets. Fig. 3 shows an example correlation output with the detected eye centered at coordinates (40,54). For all experiments presented in this paper the same UMACE eye filter was utilized.

3.3.1 Eye Location Perturbations

An issue with correlation based eye detectors is that they will also show a high response to eyebrows, nostrils, dark rimmed glasses, and strong lighting such as glare from eye glasses, and return these points as the coordinates of the eye [20, 2]. Through our analysis of the problem we have discovered that when an invalid location has the highest correlation peak value, a second or third correlation peak with a value slightly less than the highest peak is usually the true location of the eye. Therefore, we have improved upon prior correlation approaches [18, 20] to search for multiple correlation peaks on each side of the face and then determine which correlation peak is the true location of the eye. The first step in this process is to threshold the initial correlation output at 80% of the maximum value to eliminate all but the salient structures in the correlation output. Next, a unique label is assigned to each structure using connected component labeling [21]. The location of the maximum peak within each label is then located and returned as a possible eye location. This process is repeated for both sides of the face.

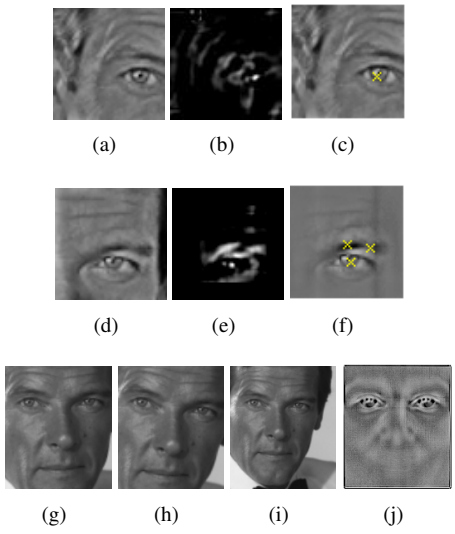


Figure 4. (a) Cropped left eye area (b) Correlation output (c) Cropped left eye area with one initial eye location returned (d) Cropped right eye area (e) Correlation output (f) Cropped right eye area with three initial eye locations returned (g-i) Image perturbations using initial left and right eye locations (j) "Average Face". The two images with the highest similarity are returned as potential eye coordinates and then sent to the facial alignment test.

The typical procedure at this point is to determine the location of the left and right eyes and then send the input image and the eye locations to a geometric normalization algorithm for processing just before face recognition. However, we are taking a different approach by sending all of the initial eye locations to the geometric normalization algorithm [1] and then selecting the "best" geometrically normalized image from all of the normalized images. Geometric normalization is a vital step in our face recognition pipeline since it reduces the variation between gallery and probe images. The geometrically normalized image is of uniform size and if the input eye coordinates are correct the output image will contain a face chip with uniform orientation. All of the geometrically normalized images are compared against a UMACE based "average" face filter using frequency based cross correlation. Our "average" face was formed by first geometrically normalizing all of the faces from the FERET [12] dataset. A UMACE filter was then synthesized from all of the normalized images. After the cross correlation operation is performed only a small region around the center of the image is searched for a global maximum. The top two left and right (x, y) eye coordinates corresponding to the image with the highest similarity are returned as potential eye coordinates and then sent to the facial alignment test. A summary of this new algorithm is shown in Fig. 4.

3.3.2 Facial Alignment Test

Once the eye perturbation algorithm has completed the top two images will be returned as input into the facial alignment test. The purpose of the test is to eliminate face images that exhibit any rotation, such as in Fig. 5(a). The eye perturbation algorithm will most often return the un-rotated face, however, it is possible to receive a greater correlation score between the rotated image and the average face UMACE filter. Before the facial image is split in half it is preprocessed by the an LBP-like normalization operator [17]. Next, the face image is split in half along the vertical axis. The right half of the face is then flipped from left to right. Normalized cross-correlation is performed between the two halves. A symmetrical facial image will exhibit a sharp peak around the middle of the correlation plane. To determine the most symmetrical image we are using the peak-to-sidelobe (PSR) ratio [19]. The PSR is a measure of the peak sharpness and is defined by Eq. 7:

$$PSR = \frac{peak - mean(sidelobe)}{\sigma(sidelobe)} \tag{7}$$

The image that produces the greatest PSR is then chosen as the input to our recognition algorithm. The facial alignment test will also function properly on off-pose facial images up to 20 degrees.

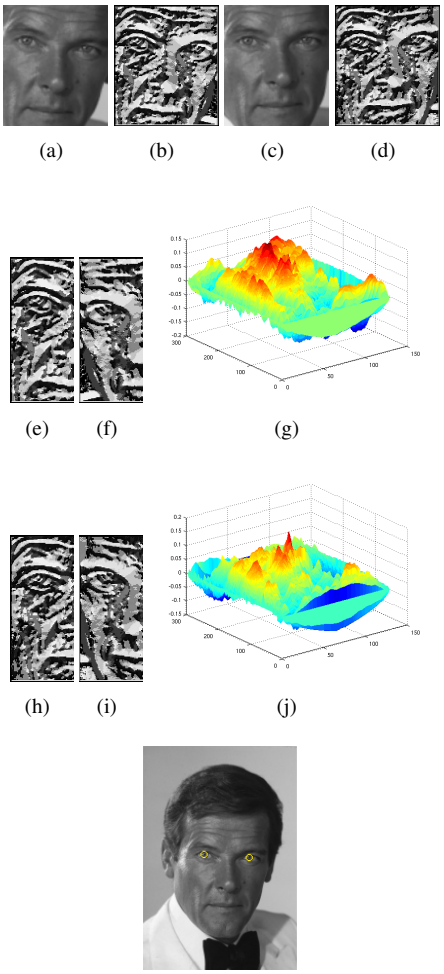


Figure 5. (a) Output face image 1 from eye perturbation algorithm (b) Face image 1 processed by an LBP-like operator (c) Output face image 2 from eye perturbation algorithm (d) Face image 2 processed by an LBP-like operator (e-f, h-i) The two images split along the vertical axis with the right half flipped from left to right (g) Normalized cross-correlation output from image 1 (j) Normalized cross-correlation output from image 2 (k) Final eye coordinates returned based on top score.

4. LFW Experiments

To first validate the effectiveness of our eye detector, a subset of the Labeled Faces in the Wild (LFW) was used. The LFW subset (dubbed LFW385) contains 385 subjects and 1,540 face chips. This subset was chosen by selecting subjects with four or more images in the original dataset. The first three images, determined by an alphabetic sort, were utilized as gallery images. The fourth image was used as the probe. Due to the constraints of our SVM-based classification method (described below), a gallery of more than a single image was required.

We then performed a series of experiments. In the first experiment, we compared detected eye positions versus manually labeled ground truth positions. The performance of our eye detection pipeline was evaluated, as well as that of our previous approach [20], which this work builds from. We define the eye localization error as the distance in pixels between the detected eye positions and ground truth eye positions. In the second experiment, we quantified the performance of our improved eye detection algorithm based on its effect on face recognition accuracy.

The recognition technique utilized is an augmented form of the V1-like features technique described by Pinto et al. [14]. Each gallery image is first filtered by an array of 96 Gabor filters, generating a large array of feature vectors. PCA is used to reduce the dimensionality of these feature vectors prior to using them to train a multiclass SVM. Due to the nature of this method of classification, several gallery images were used for each class so as to increase the accuracy of the SVM's convergence. In the model of Pinto et al., the probe images are treated the exact same way, with each resulting feature vector classified by the trained SVM. This algorithm was chosen for its relative simplicity and excellent baseline performance on popular data sets.

4.1. Eye Detection Accuracy on LFW

The first experiment incorporated all of the images from the LFW385 dataset. LFW is an unconstrained dataset by its very nature, however, the proposed eye detection approach showed excellent gains in performance for the difficult imagery as compared to our original technique. [20]. The results for eye detection are shown in Fig. 6. On the plot, the x axis represents the pixel tolerance as a function of distance from the ground truth for automatic detection, and the y axis represents the detection percentage at each tolerance.

4.2. Face Recognition Experiments on LFW

To further confirm the utility of our eye detector, we also applied it directly to face recognition, using identification accuracy to judge its performance. The facial recognition results produced by the V1-like features algorithm are presented in Table 1. Again, both the detector we propose here and that of our previous system [20] were evaluated. Our enhanced eye detector shows a performance increase of 8.36% when compared to that of our previous system [20]. The ground truth column represents the recognition results for ground truth eye coordinates. While the rank-1 recognition rates appear to be low, it must emphasized that we are evaluating *identification*, as opposed to verification. Thus, the problem is considerably harder: chance for identification is 1 in N, where N is the number of subjects in the gallery. As LFW is considered to be one of the most difficult sets for face recognition in general, it is expected that

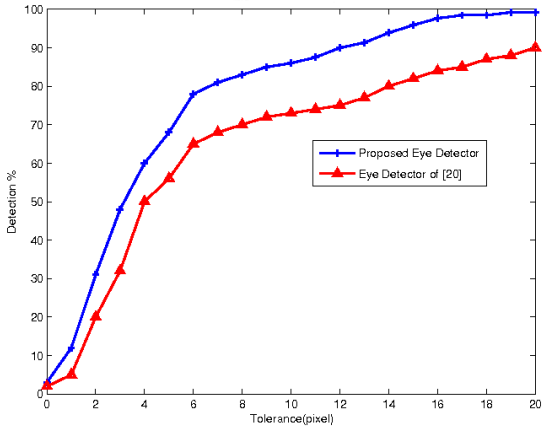


Figure 6. Accuracy of the proposed eye detection approach on the LFW385 set. Results for our previous approach [20] are also plotted. The proposed approach shows a significant increase in accuracy when compared to our previous system.

Table 1. Rank-1 recognition results for the LFW385 set produced using our proposed eye detector and that of our previous system [20]. The recognition algorithm is V1-like features [14].

Dataset	Proposed Eye Detector	Eye Detector of [20]	Ground Truth
LFW385	39.48%	31.12%	47.28%

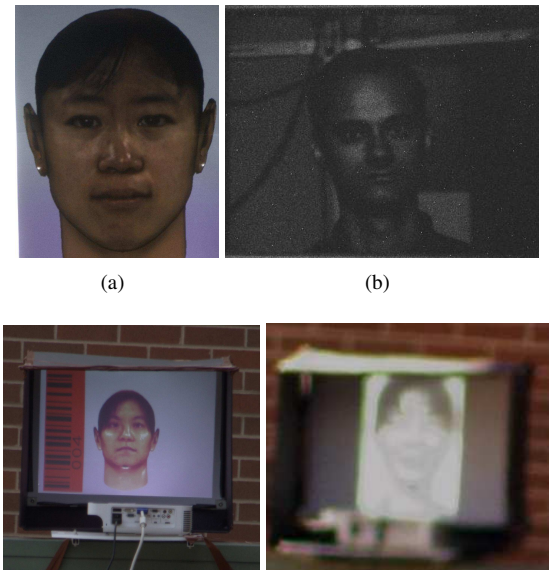
recognition rates will be considerably lower than those obtained on other, more constrained, datasets. The numbers presented are, in fact, demonstrative of a marked performance increase due to the addition of SQI normalization, skin detection, eye perturbations, and the alignment test to the correlation approach. For this subset of LFW, the recognition rate is very sensitive to detected eyes – even an error of a few pixels causes many subjects to not be recognized.

5. FDHD Experiments

Our eye detector was also tested on data from the Face Detection on Hard Datasets Competition 2011[1] (FDHD) [8]. Until this competition, there had not been a quantitative comparison of how well eye detectors perform under difficult circumstances common in surveillance applications. The organizers of the competition created a dataset of low light and long distance images that emphasize some of the problems automatic detectors encounter in surveillance scenarios. By challenging the community in this way, the FDHD competition has helped identify state-of-the-art algorithms suitable for real-world eye detection and localization. The dataset is composed of re-imaged 2D photos of faces and semi-synthetic 3D head models captured under varying conditions of lighting (including very low light), atmospheric blur, and distances of 3m, 50m, 80m, and 200m.

[1] http://vast.uccs.edu/FDHD

Example images from the dataset are shown in Fig. 7. FDHD has a defined protocol of four testing sets, each with



(a) (b) (c) (d)

Figure 7. Sample Images from the FDHD Dataset (a) 80m-500px (b) Dark-150px (c) 200m-300px (d) 200m-50px

200 randomly selected images from the master set. For the eye detection localization accuracy measurement, the error metric is defined as the Euclidean distance between each ground truth eye coordinate and the identified eye coordinate. The scores are presented using a "localization-error threshold" (LET) graph, which describes the performance of each algorithm in terms of the number of images that would be detected given a desired distance threshold. Eye detection results using this protocol are presented in Fig. 8 for our eye detector, as well as for the 11 other participants in the challenge. From Fig. 8(d) it can be seen that many eye detectors performed well on the low light imagery, with our eye detector slightly outperforming all of the other algorithms. From the results of Fig. 8 (a-c) it can be seen that our eye detector clearly outperforms all contestants, including a leading commercial SDK from Pittsburgh Pattern Recognition, in the long distance challenges.

6. Conclusion

As face recognition moves forward into real world applications, the accuracy of deployed systems becomes a primary concern. But before we can even attempt to recognize a face, we often need to perform some necessary preprocessing steps, including geometric normalization and facial feature localization, with the eyes providing the necessary reference points. Thus, in this paper, we have concentrated on the eye detection problem and techniques to improve detection accuracy. First, we presented a technique to increase the accuracy of a UMACE filter based eye detector using SQI normalization, skin detection, eye perturbations and a facial alignment test.

To validate our new approach, we performed a thorough series of experiments on data generated from one unconstrained dataset (LFW) and one extremely challenging dataset designed specifically for the evaluation of eye detectors (FDHD). Overall, by using eye perturbations and facial alignment, we can use multiple eye estimates to ultimately help select the real eye locations. The results on the LFW dataset showed that our approach achieves a significant increase in detection and recognition rates as a direct result of our improvements to our previous correlation based approach [20]. Finally, the results on the FDHD dataset showed that while many eye detection algorithms perform poorly on realistic surveillance oriented data, our algorithm did not. We outperformed all contestants, four commercial and seven non-commercial eye detection algorithms, in all categories without any per set modifications to the detection or preprocessing algorithms.

References

[1] D. Bolme, R. Beveridge, M. Teixeira, and B. Draper. The CSU Face Identification Evaluation System: Its Purpose, Features and Structure. In *ICVS*, pages 304–311, 2003. 197

[2] D. Bolme, B. Draper, and J. Beveridge. Average of Synthetic Exact Filters. In *IEEE CVPR*, pages 2105–2112, 2009. 194, 196

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 195

[4] R. Brunelli and T. Poggio. Template Matching: Matched Spatial Filters and Beyond. *Pattern Recognition*, 30:751–768, 1997. 194

[5] T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. S. Huang. Total Variation Models for Variable Lighting Face Recognition. *IEEE T-PAMI*, 28(9):1519–1524, 2006. 194

[6] G.-C. Feng and P. C. Yuen. Multi-cues eye detection on gray intensity image. *Pattern Recognition*, 34(5):1033–1046, 2001. 194

[7] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 193

[8] J. Parris et al. Face and Eye Detection on Hard Datasets. Preprint of IJCB 2011 2011, available at http://vast.uccs.edu/~tboult/PAPERS/IJCB11-Parris-et-al-FDHD.pdf. 193, 198

[9] L. Jin, X. Yuan, S. Satoh, J. Li, and L. Xia. A Hybrid Classifier for Precise and Robust Eye Detection. In *ICPR*, pages 731–735, 2006. 194

[10] B. Leite, E. Pereira, H. Gomes, L. Veloso, C. Santos, and J. Carvalho. A Learning-based Eye Detector Coupled with Eye Candidate Filtering and PCA Features. In *SIBGRAPI*, 2007. 194

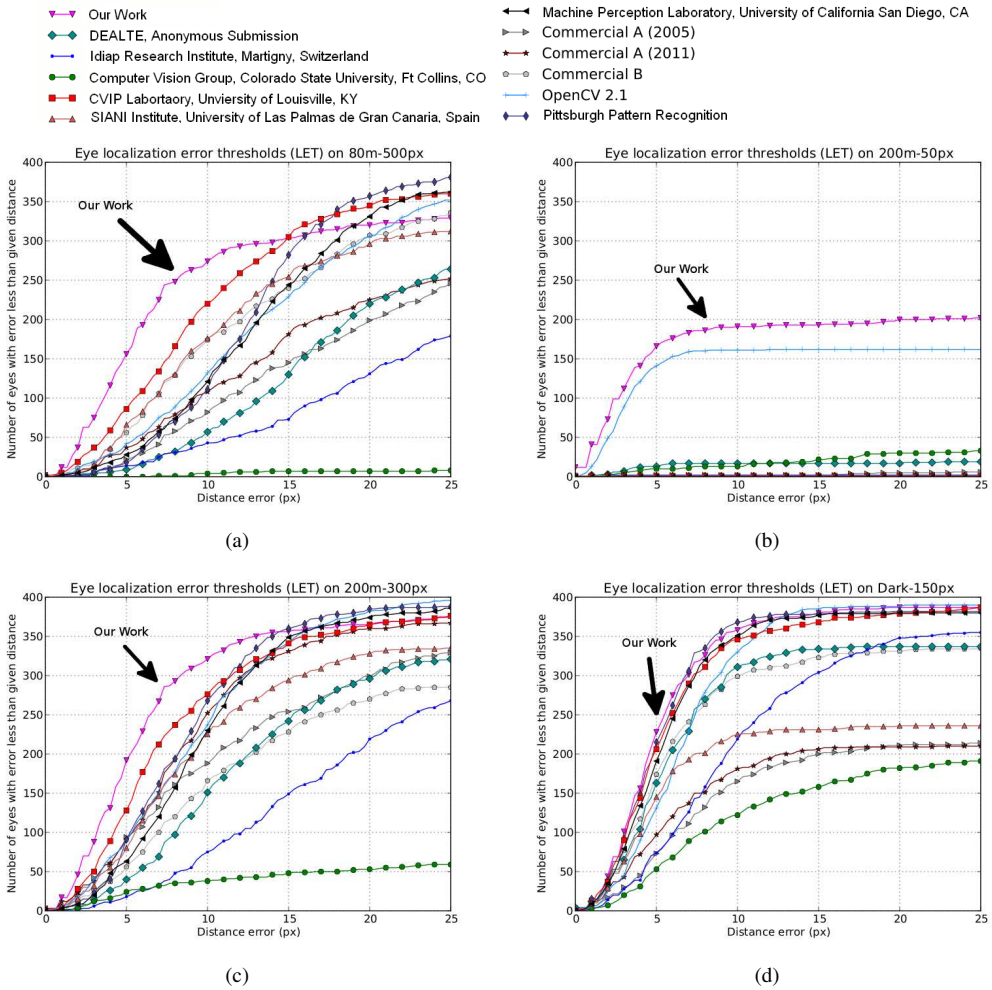


Figure 8. Eye Localization Error Threshold (LET) curves for FDHD data

[11] A. Mahalanobis, B. V. K. V. Kumar, and D. Casasent. Minimum Average Correlation Energy Filters. *Appl. Opt.*, 26(17):3633–3640, 1987. 196

[12] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The FERET Evaluation Methodology for Face Recognition Algorithms. *IEEE T-PAMI*, 22(10), 2000. 193, 196, 197

[13] J.-S. Pierrard and T. Vetter. Skin Detail Analysis for Face Recognition. In *IEEE CVPR*, pages 1–8, 2007. 195

[14] N. Pinto, J. J. DiCarlo, and D. D. Cox. How Far can you get with a Modern Face Recognition Test Set Using Only Simple Features? In *IEEE CVPR*, 2009. 198

[15] T. Riopka and T. Boult. The Eyes Have It. In *ACM SIGMM Multimedia Biometrics Methods and Applications Workshop*, pages 9–16, 2003. 193

[16] RNCOS. Global Biometric Forecast to 2012, 2011. 193

[17] A. Sapkota, B. Parks, W. Scheirer, and T. Boult. FACEGRAB: Face Recognition with General Region Assigned to Binary Operator. In *IEEE Computer Society Workshop on Biometrics*, pages 82–89, 2010. 197

[18] M. Savvides and B. V. Kumar. Efficient Design of Advanced Correlation Filters for Robust Distortion-Tolerant Face Recognition. In *IEEE AVSS*, 2003. 196

[19] M. Savvides and B. Vijaya Kumar. Efficient Design of Advanced Correlation Filters for Robust Distortion-Tolerant Face Recognition. In *IEEE AVSS*, pages 45 – 52, 2003. 197

[20] W. Scheirer, A. Rocha, B. Heflin, and T. Boult. Difficult Detection: A Comparison of Two Different Approaches to Eye Detection for Unconstrained Environments. In *IEEE BTAS*, 2009. 193, 194, 196, 198, 199

[21] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, Englewood-Cliffs NJ, 2001. 195, 196

[22] C. Shuo and C. Liu. Eye Detection Using Color Information and a New Efficient SVM. In *IEEE BTAS*, 2010. 194

[23] S. A. Sirohey and A. Rosenfeld. Eye Detection in a Face Image Using Linear and Nonlinear Filters. *Pattern Recognition*, 34(7):1367 – 1391, 2001. 194

[24] H. Wang and J. Chen. Improving Self-Quotient Image Method of NPR. *Int. Conf. on Computer Science and Software Engineering*, 6:213–216, 2008. 194

[25] P. Wang, M. B. Green, Q. Ji, and J. Wayman. Abstract Automatic Eye Detection and Its Validation. In *IEEE CVPR*, 2005. 193, 194