Automatic Eye Localization in Color Images

José Gilvan Rodrigues Maia[1], Fernando de Carvalho Gomes[1], Osvaldo de Souza[2]
[1]*Departamento de Computação – Universidade Federal do Ceará (UFC)*
[2]*Depto de Engenharia de Teleinformática – Universidade Federal do Ceará (UFC)*
60455-760 – Fortaleza – CE – Brasil
{gilvan, carvalho}@lia.ufc.br, osvaldo@lesc.ufc.br

Abstract

In this paper, we present a new efficient method for accurate eye localization in color images. Our algorithm is based on robust feature filtering and explicit geometric clustering. This combination enhances localization speed and robustness by relying on geometric relationships between pixel clusters instead of other properties extracted from the image. Furthermore, its efficiency makes it well suited for implementation in low performance devices, such as cell phones and PDAs. Experiments were conducted with 1532 face images taken from a CCD camera under (real-life) varying illumination, pose and expression conditions. The proposed method presented a localization rate of 94.125% under such circumstances.

1. Introduction

Automatic extraction of human face and facial features (eyes, nose and mouth, for example) is an essential task in various applications, including face and iris recognition, security, surveillance systems and human computer interfacing [1, 2, 3, 4, 5]. Facial feature detection and localization (FFDL) is a very important problem to be solved, because it provides meaningful input for most face processing algorithms. However, it is a computationally difficult task due to the myriad of illumination, pose, and expression possible combinations.

A general agreement is found in FFDL literature, pointing out that the eyes are the most important facial features [3, 4, 6], so most research effort in FFDL has been devoted to eye localization (EL). There are several reasons for this outstanding importance of the eyes over other facial features [3, 4, 6, 7]: eyes reveals information about the state of human beings; a face contains two eyes (if not occluded), so its position, scale and orientation can be estimated from the eye positions; and the appearance of eyes is less variant to face changes (than eyebrows, nose, ears and mouth, for example). Moreover, accurate EL provides means to identify all the other facial features of interest [1].

Although all research effort made in last years [1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14], EL remains an open problem due to increasing demand of accuracy and speed in eye localization [1, 3, 4]. It is well known that the behavior of face processing methods (face recognition, in particular) in real-life applications strongly depends on precise EL [1, 13]. In this context, most approaches to EL are not accurate enough or perform very poorly in terms of efficiency, especially when cascaded AdaBoost classifiers [5] are used and the eyes are successfully localized [5, 8] – the most time consuming case for this kind of methods.

In this paper, we present a solution to the EL problem which should be performed after face localization and is based on a robust feature filter and explicit geometric clustering. Moreover, our heuristic algorithm is efficient enough to be implemented in eye tracking for low performance processor devices.

This paper is organized as follows. Overview and related work is discussed in Section 2. The proposed method is detailed in Section 3 and evaluated in Section 4. Finally, conclusions are left to Section 5.

2. Overview

In this section we give a brief overview on the state-of-the art in eye localization. In the first subsection, we discuss how accuracy is measured for this problem and the importance of accurate methods. Related work is presented along the second subsection, and discussed in the third subsection.

2.1. Measuring eye localization accuracy

Eye localization accuracy is usually expressed as statistics of a given error measure (e.g., Euclidean distance), considering the true, manually annotated eye centers and the automatically estimated eye positions over a face dataset [4]. In order to make these statistics suitable for comparisons between different datasets, it is necessary to normalize the error measure based on the face scale, which can be estimated based on the eye distance (Figure 1).

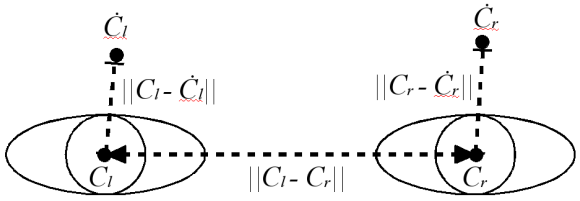

Figure 1. C_l and C_r represent exact eye centers, while the respective estimated eye centers are represented by $\bullet_l$ and $\bullet_r$. $||C_l - C_r||$ can be used to estimate accuracy.

Jesorsky et al. [3] proposed a normalized worst case error measure by using the upper bounds for estimating localization accuracy on a given face. Such metric became relatively popular, and can be expressed as

$$d_{eye} = \frac{\max\left(\left\|C_l - \dot{C}_l\right\|, \left\|C_r - \dot{C}_r\right\|\right)}{\left\|C_l - C_r\right\|}$$

EL literature points out that $d_{eye} \leq 0.25$ roughly corresponds to a distance smaller than the eye width and therefore it can be used as criterion to claim eye localization [1, 3, 4, 7]. However, this accuracy level may not be sufficient when localized positions are used for some face processing techniques, specially in the case of face recognition methods.

Campadelli et al. [1] studied the relationship between d_{eye} and the face recognition rate of baseline recognition methods and their own method. They simulated error measures and concluded that recognition rate arising from PCA, LDA and Bayesian methods strongly depends on accurate eye localization. Recognition rate on these methods rapidly decreases as the error increases, being less than *0.25* when $d_{eye} \leq 0.15$. Nevertheless, this is not surprising, since most methods are developed focusing only the face recognition task.

2.2. Related work

Jesorsky et al. [3] highlighted the strong dependency of the FR performance and the accuracy of the face alignment. They proposed both an eye localizer and a measure to evaluate quantitatively the performance of this class of methods. Their method is model-based, performing a coarse-to-fine search using the modified Hausdorff distance. The found positions are refined by applying a multi-layer perceptron trained with pupil centered images.

Ma et al. [4] proposes a three-stage method for robust and precise eye localization based on a probabilistic interpretation of the output of cascaded AdaBoost classifiers [5]. Their method works on upright frontal faces. Similarly, Tang et al. [6] combined cascaded AdaBoost and Support Vector Machine (SVM) classifiers. Their method also consists on three stages: AdaBoost face detection; AdaBoost eye detection; and a SVM post classifier that validates the reported positions.

Campadelli et al. [1] developed a SVM-based localization method that can be applied to the output of face detection methods. Their system is top-down, and relies on a SVM trained on optimally selected Haar wavelet coefficients. Eye localization is performed in two steps: eye detection, which validates the output of the face detector at the same time it provides an estimate of the eye positions; and eye localization, which refines the precision by using the specific eye pattern definition.

The method presented by Wang at al. [8] also uses the AdaBoost algorithm. The authors propose the adoption of Recursive Nonparametric Discriminant Analysis (RNDA) to overcome the limited discriminant capabilities of the Haar wavelet. RNDA is used to extract more effective features and to provide sample weights useful for the AdaBoost algorithm. As result, the classifier contains only two layers: the first has just two features – thus it is very fast, while the second has about *100* features in order to refine the search.

Titive and Bouzerdoum [9] proposed an eye detection approach based on a convolutional neural network, in which the feature extraction neurons are based on bio-physical mechanism of shunting inhibition. As each layer of the network acts as a convolution filter followed by a down-sampling operation, their method can process an entire input image and generate an output location map which is four times smaller than the original input image. As result, face detection can be performed efficiently as a real-time system.

Fu et al. [15] presented an efficient method for face detection and eye localization using neural network for color segmentation. A self-growing probabilistic decision-based neural network (SPDNN) is used to learn the conditional distribution for each color classes. Pixels of a color image are first classified into facial or non-facial regions, so that pixels in the facial region are followed by eye region segmentation. The class of each pixel is determined by using the conditional distribution of the chrominance components of pixels belonging to each class. However, skin tone detection does not perform equally well on different skin colors and is sensitive to changes in illumination [3].

The method developed by Niu et al. [10] is based on AdaBoost classifiers, while bootstrapping on both positive and negative examples. Their training procedure allows for reducing the false alarm at the same time detection rate is augmented. The authors propose two different localization procedures: one by weighting the resulting classifiers, which results on effective but time consuming detection; and the other cascading these classifiers, which stops in correspondence to the first one which detects at least one region, thus drastically reducing time consumption.

Everingham and Zisserman [11] presented and compared three approaches for eye localization: a regression method which directly minimizes the prediction error; a Bayesian approach, consisting on two distinct, independently built probabilistic models of the eye and non-eye appearance; a single strong classifier trained for eye detection using AdaBoost. All the methods are trained and tested on the same images, and during detection they are applied in cascade to a Viola-Jones face detector. Results show that the simple Bayesian model outperforms the others. The authors explain this fact by drawing attention to the difficulty of using classifiers for the task of localization.

2.3. Discussion

Eye localization methods shown in the last subsection reported acceptable performance and accuracy on images containing upright frontal faces, supporting head rotations in and out of the plane. However, it is difficult to establish a reasonable comparison of their results due to the lack of a standard error metric [3].

Most of these methods are built on top of the work by Viola and Jones [4, 6, 7, 8, 10, 11], using cascaded AdaBoost classifiers for effective localization of faces or the eyes themselves. Here are some important observations about this kind of methods:

- multi-scale detection of faces affects performance, since the input image has to be scanned multiple times at different scales. This is also valid for classifiers trained with fixed scale;
- eye localization can be performed more efficiently, because a rough estimate of scale is available when a face is found, making it possible to dramatically reduce the number of iterations over eye scales.
- the same integral image used for finding faces can be reused for this task, in the case a Viola and Jones detector is used for eye localization;
- assuming upright faces, each eye lies in a sub-region that is roughly four times smaller than the reported face's estimated size – by searching the left (right) eye only in the upper left (right) part of the face region;
- EL accuracy depends on the scanning algorithm. Errors can arise due to iteration over different scales, and the merging process when multiple detections are reported, among other situations [5, 8];

In general, eye localization methods consist on classifiers built using a combination of well-known approaches for pattern recognition and machine learning: neural networks, SVM, boosting, regression, Bayesian modeling, among others. We propose a new eye localization method that is based on a simpler approach and exploits visual clues present in color images. Multi-scale localization is naturally supported due to the geometric nature of the clustering algorithm used as part of the search.

3. CLUED: clustering-based eye detection

Our method is integrated into an eye localization system consisting on two phases: face detection and eye localization. The system is depicted by Figure 2. We build on top of the work by Viola and Jones [5], and rely on their method for detecting upright frontal faces before starting eye localization. Actually, any method providing a reliable estimate of face position and size can be used at this first phase.

Eye localization consists on three steps, which are explained in detail along the next subsections: feature segmentation; point clustering; and eye localization. In the first step, face region is segmented in order to obtain candidate feature points. The resulting points are then grouped in clusters in the second step. Finally, in the third step, clusters representing the eyes are selected and the eye positions are determined.

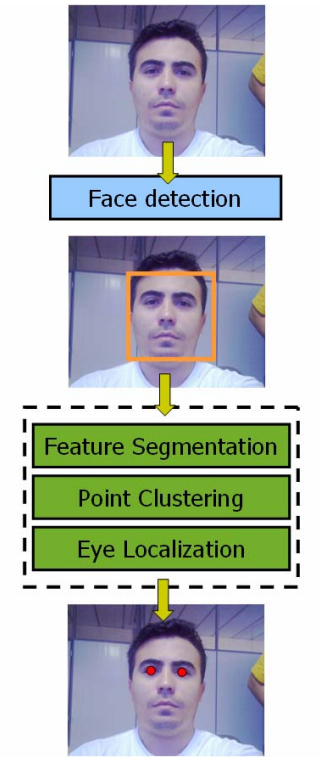

Figure 2. Overview of our eye localization system. First, face position and scale is estimated using a reliable method, and then eye localization (denoted by the dotted box) is performed.

3.1. Feature segmentation filter

In this step, we want to select candidate points representing the eyes. This step is carried out using the *rg*-normalized color space, which is derived from the RGB color space by normalizing (dividing) each component by the sum of the red, green and blue intensities.

This color space has two important properties for this purpose: the dependency of r and g on the brightness is greatly diminished after normalization; it is relatively invariant to changes of surface orientation relative to light sources [12]. Due to this, *rg*-normalized color space was applied in human skin segmentation [13].

For developing our feature filter, first we observe that human faces consist in two parts: skin, mainly on the forehead, cheeks and nose; and facial features, such as the eyes, eyebrows and nostrils. Due to nature of skin, it is possible to roughly distinguish between skin and these features by considering only the r_n component in the *rg*-normalized space, as shown in Figure 3.

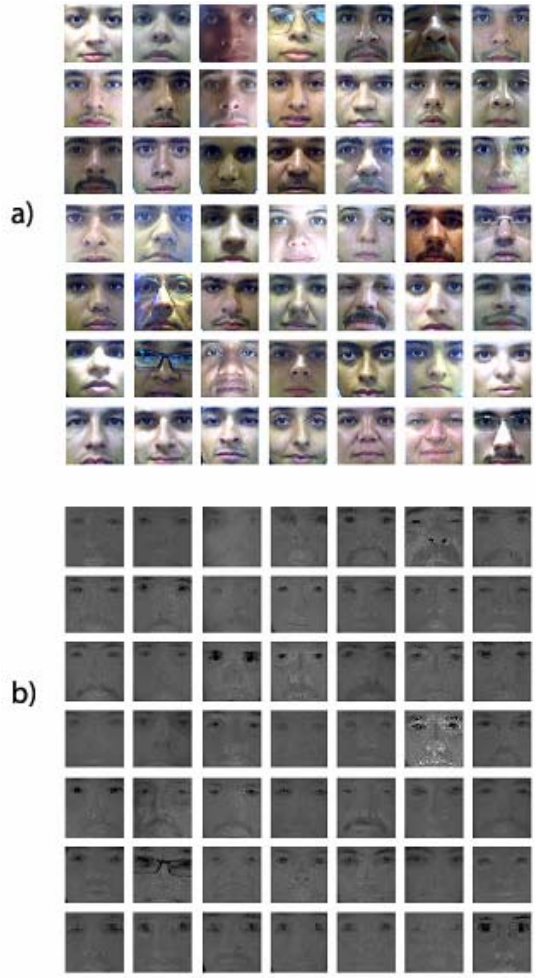

Figure 3. Color face images (a) representing different conditions of face acquisition found in real-life (skin color, illumination, eyeglasses, noise), and each corresponding r_n image (b) highlighting features, specially the eyes. Faces were aligned and rescaled from their original poses for presentation purposes.

Let us define an RGB image I as a function mapping two integers, representing a given pixel, into the corresponding red, green and blue components denoted by I_r, I_g and I_b, respectively

$$I(x,y) \rightarrow (I_r(x,y), I_g(x,y), I_b(x,y)),$$

so that

$$I_i(x,y) \in [0, 2^{c_i} - 1]$$

where c_i denotes bit depth in the channel i (r, g, or b).

For convenience, the conversion from I to r_n was slightly modified by adding 1 to the denominator in order to avoid divisions by zero. Moreover, we

consider the complement of r_n, since facial features tend to dark in this color space. Such conversion is defined as

$$R_I(x,y) = \left[1 - \frac{I_r(x,y)}{1 + I_r(x,y) + I_g(x,y) + I_b(x,y)} \right]$$

Once the conversion is done, histogram equalization is performed over R_I in order to enhance robustness of our feature segmentation, resulting in a monochromatic image

$$ER_I = HistogramEqualize(R_I).$$

Finally, a threshold is applied in order to segment facial features, resulting in a binary image T_I, defined as

$$T_I = \begin{cases} 1, & ER_I(x,y) > t \\ 0, & otherwise \end{cases}$$

Noise and other features such as the mouth, nostrils and eyebrows may appear in T_I (see Figure 4). However, the goal of this step is to find candidate eye regions used in the subsequent steps.

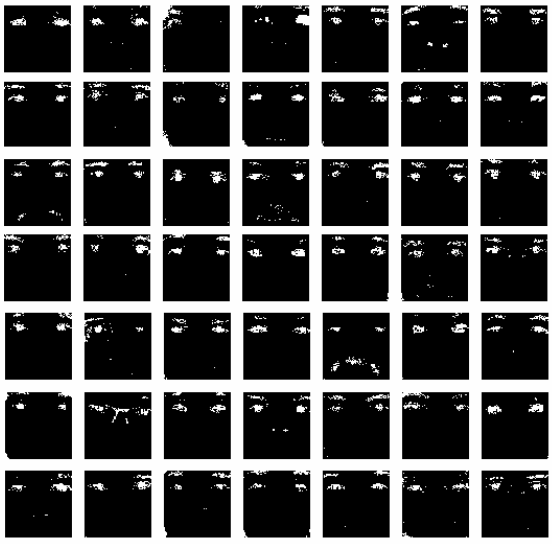

Figure 4. Result of feature segmentation over faces from Figure 3-a. Eye regions are highlighted by using *t=0.96*, and only one eye was lost (*2%* error in this case). Segmentation also produces "artifacts" arising from noise, nostrils, eyebrows, eyeglasses and beard.

3.2. Efficient point clustering

At this stage, the problem of localization is solved from a geometric perspective, where candidate points are grouped into clusters for further localization. The input for the clustering algorithm is expressed as the set of all candidate feature points resulting from the feature segmentation

$$P_I = \left\{ \arg(T_I(x,y)) \middle| T_I(x,y) = 1 \right\}.$$

Contrary to Armanag et al. [14], which proposed a color tone-based clustering algorithm for correcting their Bayesian eye classifier, we employ a purely geometric clustering method. More specifically, we adapted the k-means algorithm [16] to our problem. Given the point set P_I and the desired number n_c of clusters to compute, the clustering method iteratively moves the (randomly placed at startup) cluster centers q_j by selecting the nearest points in P_I. Each cluster center q_j is then recomputed before the next iteration takes place, by selecting the mean of the points nearest to q_j. This is illustrated by Figure 5.

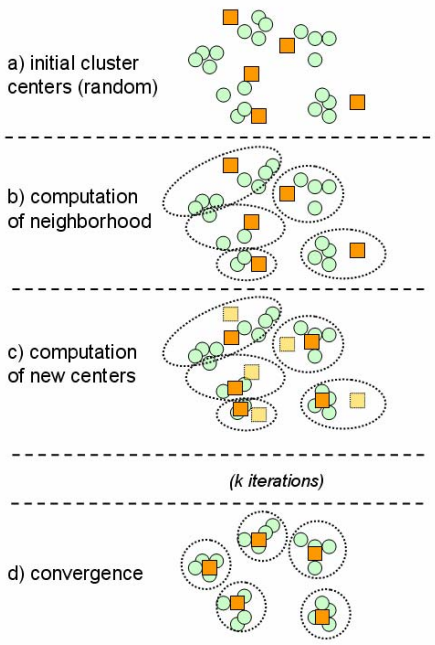

Figure 5. Clustering algorithm, as proposed by MacQueen [16]. Initial cluster centers are randomly chosen at the start (a). The nearest neighborhood is computed (b), and then each cluster q_i center is updated to the mean of its neighborhood before the next iteration is executed.

The method stops when cluster centers stop moving (thus convergence is found), or when points are

moving from one cluster to another. Actually, this method presents very fast convergence [15], so only k iterations are usually allowed. Our experiments show that 5 iterations are sufficient for convergence when using up to 36 clusters. Final result of clustering is depicted in Figure 6, in which clustering considers candidate points reported from the entire face for explanation purposes. Actually, our algorithm assumes face detection has taken place and only the upper part of the face region is processed.

Figure 6. Result of feature segmentation, followed by point clustering using $t=0.96$, $k=5$, $n_c=8$.

In addition, we build a kd-tree considering the current position of cluster centers per iteration. This spatial data structure is built in $O(n_c log(n_c))$ and allows to find the closest cluster center for a given point in $O(log(n_c))$. Because of this, the clustering step performs in $O(k(p+n_c)log(n_c))$, where p denotes the number of candidate feature points for a given face. As n_c is much smaller than p and k is constant, the clustering algorithm complexity is $O((p)log(n_c))$. Any spatial acceleration data structure, such as a Quadtree, can be used to speedup the search for nearest cluster center. We used a kd-tree because it is very compact and more flexible to build than a Quadtree.

Once point clustering is finished, finding the eyes is a relatively simple problem to solve. As preprocessing, the face scale estimate is used to remove very large and very small clusters, representing noise, other features (nostrils, eyebrows), eyeglasses and hair. By assuming that clustering only considers the upper half of the image, the eyes are represented by two clusters: one in the left upper part (the left eye); and the other in the right upper part (the right eye). Moreover, it is assumed that eye localization fails when no cluster is present in a given side of the image.

After this point, the problem of localization is solved by choosing an adequate cluster to represent each eye. This is performed by iteratively scanning the few resulting cluster points for the greatest ball in each side of the image that is closest to the image center, while constraining the in-plane rotation to a maximum of $10°$. Figure 7 illustrates the results corresponding to images presented in Figure 3-a.

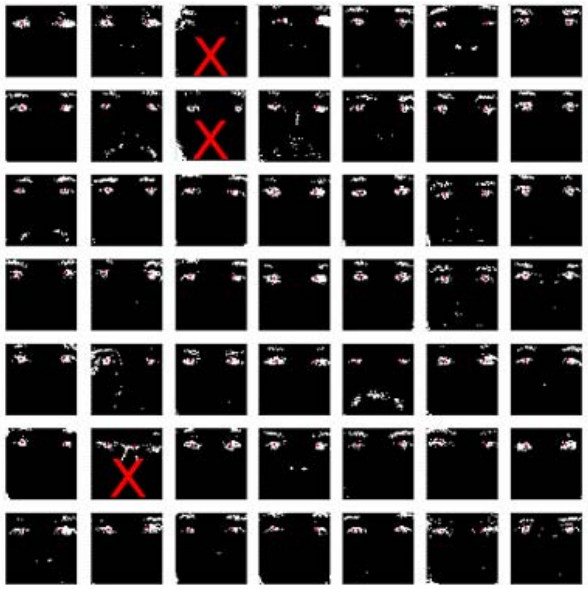

Figure 7. Eye localization based on point clustering using $t=0.96$, $k=5$, $n_c=8$. Our simple method missed only 2 eye pairs from 49 faces in Figure 3a, and only reported 1 wrong eye pair – when the user is wearing thick eyeglasses. In this case, localization rate is *93.87%* and average d_{eye} is less than *0.085*.

4. Evaluation and discussion

The error rate criterion proposed by Jesorsky et al. [3] was adopted in order to evaluate the accuracy of our eye localization method, since this measure provides means for comparing accuracy between different face databases. Experiments were conducted with 1532 face images taken from 61 people using a CCD camera under (real-life) varying conditions of illumination, pose and accessories. CLUED presented a localization rate of *94.125%* under there circumstances. Moreover, it presented $average(d_{eye})<0.091$, which is suitable for most face processing techniques since an error of *0.25* is the lower bound in literature for claiming eye localization [1, 3, 4, 7]. Our method also compares favorably against well-known EL algorithms operating over face detection algorithms [1, 3] for $d_{eye} < 0.1$, as shown on Table 1.

From the perspective of efficiency, our method provides a fast solution for eye detection, allowing for real-time performance. In particular, a kd-tree is used for cluster center updates during the clustering step, thus performing faster than an exhaustive k-means implementation. Furthermore, the clustering step allows for multi-scale eye detection without requiring any additional effort. Because of its efficiency, our method was successfully implemented in low

performance embedded devices: acceptable speed and accuracy were also presented in this situation – about 5 frames per second on QVGA images.

Table 1. Eye localization rate when $d_{eye} < 0.1$. The real localization rate in this case is not clear in [3], so we assumed a safe upper bound.

EL method	Face database	EL rate for $d_{eye} < 0.1$
CLUED	custom	92.6%
Campadelli et al.	BioID	83.8%
Campadelli et al.	XM2VTS	95.9%
Jesorsky et al.	BioID	about 93%
Jesorsky et al.	XM2VTS	about 80%

Our algorithm is simple and supports head rotations up to *10°* in and out of the plane. Moreover, this upright frontal face restriction is not intrusive and it is adequate for almost any real-life systems assuming user cooperation. As experiments show, accessories can affect EL rate and accuracy. However, our simple algorithm performed reasonably well even in this situation (about *75%* of EL).

A better feature filtering technique may improve EL rate and accuracy, because the technique proposed in this paper may not work properly in adverse illumination situations, as occurs for third column, first row in Figure 7. Other cluster selection criterion may also yield better results, since fails may occur for simple or more difficult situations when the user is using accessories. Such cases are depicted by faces placed at the third column, second row; and the second column, sixth row in Figure 7, respectively.

5. Conclusions

Automatic, accurate eye localization is a very important problem in biometrics because it provides meaningful input for most face processing algorithms. Moreover, all other face features of interest can be identified based on the eye positions. Most methods found in EL literature are inaccurate or perform poorly in terms of speed, thus affecting the behavior of real-life face processing applications.

In this work, we present an efficient solution to the EL problem, which should take place after face detection. Our method is based on a robust feature filter and explicit geometric clustering, allowing for robust and precise eye localization. Moreover, our heuristic algorithm is efficient enough to be implemented in eye tracking systems for low performance processor devices.

The proposed method presents a localization rate of *94.125%* under the minimal requirements for eye localization is met, and compares favorably against well-known EL algorithms operating over face detection algorithms under precise localization situations. Future work includes adapting our technique in order to handle grayscale images.

6. Acknowledgements

This work was sponsored by Samsung Brazil under Law 8248 contract. The author José Gilvan Rodrigues Maia is funded by FUNCAP.

7. References

[1] P. Campadelli, R. Lanzarotti, and G. Lipori, "Eye localization for face recognition". RAIRO Theoretical Informatics and Applications no. 40, 2006, pp. 123-139.

[2] C. Huang, H. Ai, Y. Li, and S. Lao, "High-Performance Rotation Invariant Multiview Face Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, no. 4, April 2007, pp. 671-686.

[3] O. Jesorsky, K. J., Kirchberg, and R. W., "Robust face detection using the hausdorff distance". Lecture Notes in Computer Science, 2001, 2091:212 – 227.

[4] Y. Ma, X. Ding, Z. Wang, and N. Wang, "Robust precise eye location under probabilistic framework". Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition, 2004.

[5] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features". Proc. IEEE Conf. On Computer Vision and Pattern Recognition, 2001.

[6] X. Tang, Z. Ou, T. Su, H. Sun, and P. Zhao. "Robust Precise Eye Location by AdaBoost and SVM Techniques". Proc. Int'l Symposium on Neural Networks, pages 93–98, 2005.

[7] Z.H. Zhou, and X. Geng, "Projection functions for eye detection". Pattern Recognition Journal, 37:1049–1056, 2004.

[8] P. Wang, M. Green, Q. Ji, and J. Waymanm, "Automatic eye detection and its validation". Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2005.

[9] F.H., Tivive, and A. Bouzerdoum, "A Fast Neural-Based Eye Detection System". Proc. of IEEE International Symposium on Intelligent Signal Processing and Communication Systems, 13-16, 2005, pp. 641-644.

[10] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao, "2D Cascaded AdaBoost for Eye Localization". Proc. Of the 18th International Conference on Pattern Recognition, 2006.

[11] M. R., Everingham, and A. Zisserman, "Regression and classification approaches to eye localization in face images".

Proc. of the 7th International Conference on Automatic Face and Gesture Recognition, 2006.

[12] C. P. Papageordiou, M. Oren, and T. Poggio, “A general framework for object detection”. Proceedings of IEEE International Conference on Computer Vision, 1998, pp. 555-562.

[13] Li, S. Z., and Jain, A. K. (Editors). *Handbook of Face Recognition*. Springer, New York, 2005.

[14] S. Amarnag, R. S. Kumaran, and J. N. Gowdy, “Real time eye tracking for human computer interfaces”. Proc. of the International Conference on Multimedia and Expo - Volume 3, 2003, pp. 557-560.

[15] H.-C, Fu, P.S., Lai, R.S., Lou, R.S., and H.-T, Pao, “Face detection and eye localization by neural network based color segmentation”. Proceedings of the 2000 IEEE Signal Processing Society Workshop Volume 2, Issue , 2000 Page(s):507 - 516 vol.2.

[16] J. B. MacQueen (1967), "Some Methods for classification and Analysis of Multivariate Observations”. Proc. 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297.

[17] BioID face database (free). Available at *http://www.humanscan.de/support/downloads/facedb.php*.

[18] XM2VTS face database (commercial). Available at *http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/*.