Real-time eye pupil localization using Hough regression forest

Amine Kacete, Renaud Séguier, Jérôme Royan, Michel Collobert, Catherine Soladie

Real-time eye pupil localization using Hough regression forest

Amine Kacete
IRT B-com
amine.kacete@b-com.com

Renaud Séguier
IRT B-com
renaud.seguier@b-com.com

Jérôme Royan
IRT B-com
jerome.royan@b-com.com

Michel Collobert
IRT B-com
michel.collobert@b-com.com

Catherine Soladie
CentraleSupelec
catherine.soladie@centralesupelec.fr

Abstract

Eyes are one of the most salient features of the human face, and the location of the pupil allows access to important information which can be used in several computer vision applications. Several commercial eye-trackers can estimate with good accuracy the pupil location, but need complex hardware specifications and a controlled user environment (high eye image resolution, good illumination, small head pose variations) making these solutions difficult to use in an arbitrary environment. In this paper, we present an approach based on Hough randomized regression trees. We demonstrate, by several evaluations on challenging public datasets that our approach is very robust to illumination, scale, eye movements and high head pose variations and yields a significant improvement compared to a wide range of state-of-the-art methods..

1. Introduction

Pupil location plays a key role in several computer vision applications especially in face analysis. In facial expression recognition fields, it allows access to important information such as the cognitive and expressive state of the person. In biometric applications, face identification and recognition are closely linked to the pupil. Typically, HCI applications principally use this characteristic.

Several industrial solutions are commercialized. They provide good accuracy on pupil location. Some of these solutions use complex hardware specifications (embedded camera on a head-mounted system) making them inappropriate for large scale public use. Other solutions use a range of infrared cameras to detect corneal reflection and estimate the pupil position, but they remain very sensitive to illumination conditions. Hansen et al give a detailed survey of pupil location methods from a single monocular camera in [11].

Figure 1. Real-time 2D eye-pupil estimation by our approach.

In this paper, we present a non-intrusive robust pupil localization from a simple uncalibrated monocular camera. Based on an ensemble of randomized regression trees grouped in a single forest, our method detects 2D pupil location in real-time as illustrated in Figure 1. The forest learns the spatial relations between images patches and pupil 2D location from a training set. The training set is chosen so as to cover all possible variations relative to the pupil appearances. We derive a function that estimates the final 2D pupil location in the image space from the projected forest outputs on Hough space in a pyramidal way by merging.

In our experiments, we evaluate the robustness and accuracy of our method on challenging databases. We demonstrate that the obtained results are comparable and sometimes superior to a wide range the state-of-the-art methods.

2. Related work

Our work relates to a large set of existing methods trying to detect the pupil location from monocular camera with sufficient accuracy. [17, 21, 22] present the most relevant methods where the main principle is to detect the face using

the Viola Jones method [24], extract rough regions around the eyes using anthropomorphic relations then estimate the spatial position of the pupil on the image space.

Timm and Barth [21] use the geometric aspect of the pupil by defining an objective function based on an image gradient that takes its maximum at the intersection of the gradient vectors. This method is very robust under illumination and scale variations. Nevertheless, with significant head pose variations, circularity of the pupil is not guaranteed giving bad estimates.

Valenti et al. [22] use an isophote curvature which represents a set of curves that connects points of equal intensity. They extract a SIFT [15] vector for each candidate location and compare it to a given template in a defined database to get the final decision. Like the method used by Timm et al., this solution suffers significantly from head pose variations since vectors pointing to the isophotes centers give a wrong estimate.

Markus et al. [17] ignore geometric assumptions, they use a machine-learning approach by training a cascaded ensembles of trees to learn the mapping between eyes images appearances and 2D pupil locations. Each ensemble processes a given scale i and represents the input of the following ensemble relative to the scale $i-1$ up to the final output (the number of scales defines the number of ensembles). Their final learning includes one hundred trees organized in five ensembles trained with six million images.

The latter method seems the most robust and accurate approach under different constraints such as low resolution, illumination conditions and head pose variations but it needs a strong initialization assumption due to the designing of their processing model and it suffers from some intra-user variations.

Our approach is relatively close to this method as we use the same learning approach, except that we integrate a generalized Hough space in the final estimation. **To handle scale variations without introducing prior information, we introduce a new voting space to better estimate the global maximum of the eye-pupil location by merging the Hough spaces resulting from each scale**. In addition, we show that with a smaller set of trees than [17] we can obtain similar, or even better results. We also show that this method can be extended to different regression problems. To the best of our knowledge, our proposal is the first approach that uses Hough regression forest for pupil localization.

3. Our method

We use randomized regression trees to estimate the 2D pupil locations from 2D images on detected faces. In section 3.1, we provide some background on regression trees, then we detail the training step of the trees in section 3.2. In 3.3, we explain the mapping from the trees outputs to the 2D pupil location using the whole ensemble of the trees. Finally, in 3.4, we focus on the training data.

3.1. Regression forest

Random forest-based techniques are increasingly used for computer vision-based applications. Introduced by Breiman [2], randomized trees deal with different tasks, in classification [8, 14, 16, 20], in regression [4, 5, 12, 17], in density estimation [1, 19] and in manifold learning [10].

Regression forest is an ensemble of tree predictors. Each tree splits the initial problem in two low complexity problems in a recursive way. This subdivision is performed by a simple test at each node of the tree. The tests are selected in order to achieve an optimal clustering. The terminal nodes of the tree called leaves, store the models that approximate the best desired output. To achieve a high generalization, two randomness are introduced during the training of the trees, first, in the set of training data provided for each tree, second, in the set of tests computed for optimization for each node. A simple illustration of a regression forest used in our work is represented in Figure 2.

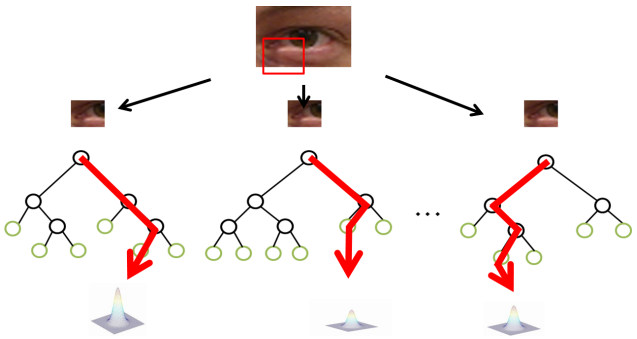


Figure 2. Overview of a regression forest. For a given input image, each tree applies at each node a simple binary test until reaching the leaf which stores the output models. By combining all the trees, the forest returns the aggregation of all the informations stored in the leaves.

3.2. Training

Each tree t in the forest $\mathcal{T}_{t=1:T}$ (T defines the forest size) is built separately from a set of annotated patches $\{w_i = f_i, c_i, y_i\}$ randomly selected from all the training examples, where:

- f_i is the extracted features vector from a given patch.
- c_i represents the class of a given patch.
- y_i represents the output variable to regress.

In our work, the extracted feature vector f_i relates to the intensities values of the patches. $c_i \in \{0, 1\}$, all the patches extracted from the pupil images are assigned to class 1 for

positive patches and class 0 represents all the patches extracted from arbitrary examples as negatives patches. y_i represents the offset vector stretching from the patch center to the pupil center in the image from which it was extracted ($y_i = 0$ for the negative patches.)

During training, we define a simple test at each node starting from the root, randomly selected from a large set of possible tests. Similar to [17] which finds its origin in [14] , the test is defined as :

$$\begin{cases} 1, & if \quad f_i(x_1) - f_i(x_2) \leq \tau \\ 0, & otherwise \end{cases}$$

where $\{f_i(x_1) - f_i(x_2)\}$ represents the difference in pixel intensities between two locations (x_1,x_2) in the patch and τ is a random threshold. Consider an ensemble of training data for a given node S, if the test is verified, the training data is sent to the right child node S^R,otherwise, it is sent to the left child node S^L. Figure 3 shows a patch extracted from a positive image with 2D offset vector and two random pixel locations x_1,x_2 for the binary test. The building

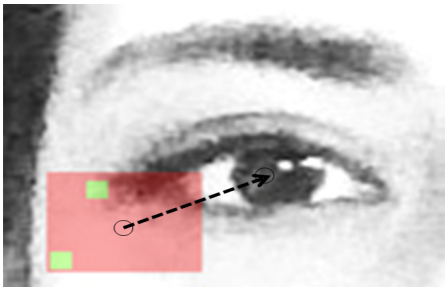


Figure 3. An example of a positive patch (represented by a red rectangle) and the associated 2D offset vector. The green rectangles represent the two selected pixel locations for the binary test.

of each tree is supervised. It consists of finding for each node during training the optimal binary test $\{x_1^*, x_2^*, \tau^*\}$ that maximizes the training data clustering defined by the purity of the two child nodes. To build trees able to capture both classification and regression information, two objective functions are evaluated such as in [6]:

- **Classification**

$$Q_0 = H(S) - \sum_{n \in \{L,R\}} \frac{|S^n|}{|S|} H(S^n) \qquad (1)$$

with

$$H(S) = - \sum_{c \in \{0,1\}} p(c/S) \ln \big(p(c/S)\big) \qquad (2)$$

Q_0 represents the information gain equal to the entropy H of the parent node S minus the weighted sum of the entropy of the left child S^L and the right child S^R.

- **Regression**

$$Q_1 = \sum_{n \in \{L,R\}} \left(\sum_{i \in S^n} ||y_i - \frac{\sum_{i \in S^n} y_i}{|S^n|}||_2^2 \right) \qquad (3)$$

Q_1 represents the sum of all the distances from each offset vector to the mean in each child node.

Maximizing Q_0 and minimizing Q_1 aims to fix the optimal binary test for each node until reaching the leaves.

Each leaf l stores the following information :

- $p(c/\{w_i\}_{i \in l})$ captures the probability of each class in the reached leaf l.
- $\mathcal{N}(y_l, \bar{y_l}, \Sigma_l^y)$ represents the Gaussian distribution of all the offset vectors reaching the leaf l. $\bar{y_l}$ and Σ_l^y represent the mean and the covariance of the offset vectors respectively .
- $\{y_i\}_{i \in l}$ represents the set of all the offset vectors reaching the leaf l.

3.3. Testing

Given an unseen image, we build an image pyramid to model the scale variation. For each scale of the image pyramid, we extract a number of fixed size patches. Each patch is passed through all the trees of the forest. Each tree tests the patch using all binary tests fixed at each node until reaching the leaf which gives the stored informations $\{p(c = 1|w), \mathcal{N}(y, \bar{y}, \Sigma^y), \{y_i\}_{i \in l}\}$.

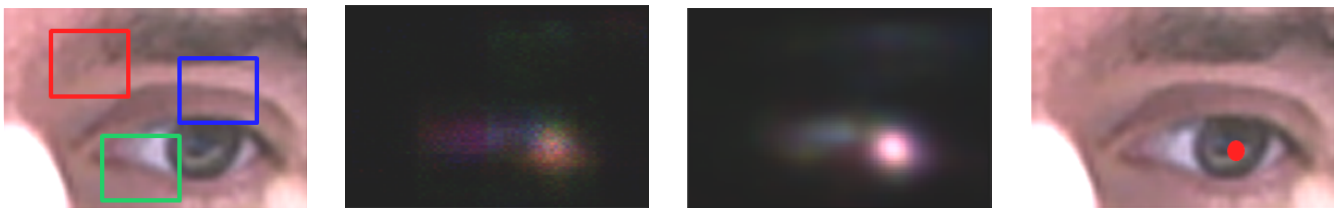
Using all the leaves returned by the forest for all the patches, we build the Hough image $\mathcal{H}_s$ for each scale from the pyramid. We project the set of pupil location candidates by adding all the offset vectors $\{y_i\}_{i \in l}$ to the patch center c. For a single tree, the candidates are represented as the sum of a Dirac weighted by the probability of belonging to the eye $p(c = 1)$ in the reached leaf. Then, we average all the outputs over the forest. For a given number Ω of patches extracted from a given image from the pyramid, the Hough image $\mathcal{H}_s$ is represented as follows :

$$\mathcal{H}_s = \sum_{w \in \Omega} \left(\sum_{t \in T} \left(\sum_{i \in l} \frac{p(c = 1|l)}{|T|.|l|} \delta(y_i + c) \right) \right) \qquad (4)$$

All the non-informative leaves presenting a high variance defined as $trace(\Sigma_l^y)$ and a low probability $p(c = 1|w)$ are discarded.

Unlike previous works as [6] where the global maximum is estimated using the best scale in the performed image pyramid, in our work, we consider all the votes casted by all the scales in the pyramid. By performing a weighted merge of the different Hough space $\mathcal{H}_s$, we build a global voting space $\mathcal{H}$ as follow: $\mathcal{H}$

$$\mathcal{H} = \sum_{s \in S} \sigma_s \mathcal{H}_s \qquad (5)$$



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

Figure 4. Hough forest voting strategy. (a) some patches extracted from the test image. (b) for each patch, the forest casts a number of votes (the color indicates the contribution of each patch). (c) aggregation of votes from all the extracted patches. (d) mapping the pick location from the Hough space into the image space.

where:

$$\sigma_s = \frac{max(\mathcal{H}_s)}{\sum_{s \in SC} max(\mathcal{H}_s)} \tag{6}$$

Figure 4 gives an overview of the building of the Hough image for a given image of the pyramid.

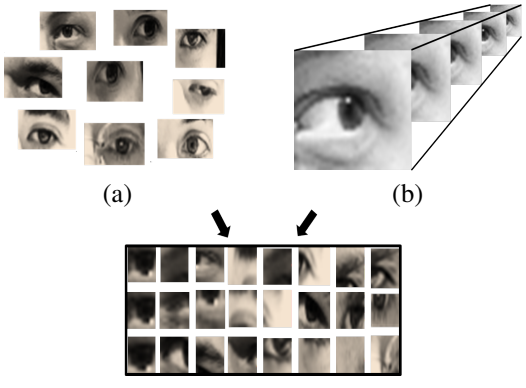
3.4. Training data

To build our forest, we use annotated datasets [23] and [25]. We perform face detection through all the images and extract rough regions around the eyes using anthropomorphic relations.

To enhance the generalization of our trees, we introduce some perturbations in the extracted regions as [17] in scale with $[+30\%, -30\%]$ and in 2D pupil location by $[-25\%, +25\%]$ from the original.

We collect $10k$ perturbed eye region samples from which we extract 50 patches of a fixed size (16×16) per sample. Thus, we obtain $500k$ positive images.

According to our problem, we extract a set of negative patches from regions belonging to the face but different to the eyes regions. Figure 5 shows some examples from the training data.



(a) (b) (c)

Figure 5. Training data generation. (a) 2D pupil location perturbations. (b) Scale perturbations. (c) Some example of final extracted patches for the forest training.

4. Experimental results

In our experiments, we build a forest of 30 trees, the last 10 trees are trained from the misclassified patches used for the training of the first 20 trees. The depth of trees is fixed at 15. As in [6], the regression and classification objective functions are selected with equal probability at each node, if the number of negative patches is reduced by more than 90%, the regression function is performed. For optimization of binary tests at each node, 10000 tests are evaluated.

During the test, the number of scales used for the pyramid images is fixed to five. We use the Viola Jones face detector [24] to extract the face, then we extract two rough regions around the eyes.

4.1. Quantitative results on still images

To compare our method to the state-of-the-art, the database BioID[1] is used. It contains 1521 annotated gray-scale images. BioID is considered among the most challenging databases for pupil localization due to its significant variations in terms of head pose variations, scale and illumination conditions. Like the majority of pupil localization algorithms, the metric introduced by [13] is used. It is defined by :

$$e = \frac{max\{D_L, D_R\}}{D} \tag{7}$$

where D_L and D_R represent the Euclidean distances from the estimated pupil locations to those in the ground truth. D is the Euclidean distance between the ground truth pupil locations.

Table. 1 shows the comparison of our method with the state-of- the-art according to equation 7. It represents the percentage of correct estimations for the given threshold(we use the same values provided by [17, 21, 22]).

- $e \leq 0.25$: Usually used for face matching, it corresponds to the distances between the pupil center and the eye corner. It indicates that the estimation belongs to the eye region which represents a low level of pre-

[1] https://www.bioid.com/About/BioID-Face-Database

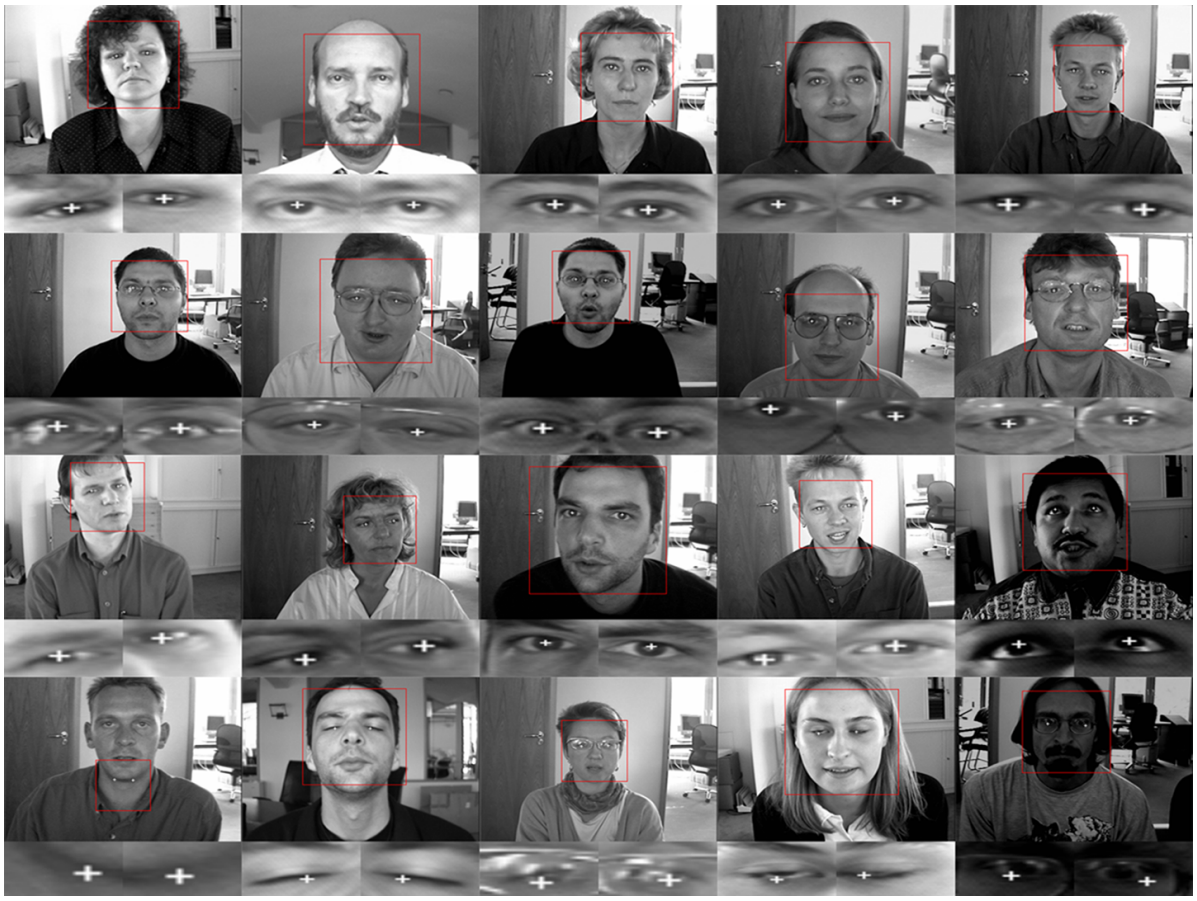


Figure 6. Some pupil estimations on the BioID database using our method. First row: good estimations on images with favorable conditions. Second and third rows: robust estimations under some unfavorable conditions (dark illumination, head pose variations and presence of glasses). Last row: some typical bad estimations using our method.

Methods	$e \leq 0.05$	$e \leq 0.10$	$e \leq 0.15$	$e \leq 0.25$
Jesorsky et al [13]	38.0	78.8	84.7	91.8
Timm et al [21]	82.5	93.4	95.2	98.0
Valenti et al. [22]	84.1	91.0	94.0	96.6
Markus et al [17]	89.9	97.1	–	99.7
Our method	**91.3**	**97.9**	**98.5**	**99.6**

Table 1. Comparison of pupil 2D localization on the BioID database. The authors of [17] do not provide the result for $e \leq 0.15$ but we point it out as an empty case.

cision. The majority of methods gives approximately the same results.

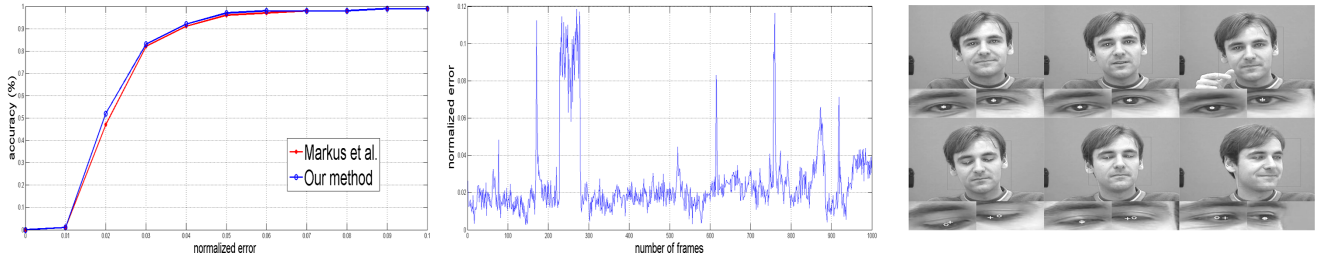
- $e \leq 0.15$ and $e \leq 0.10$: Our method yields better results compared to [22] and [21]. The circularity of the pupil which represents a strong assumption of the last two methods is not guaranteed due to significant changes in the head pose. The presence of eye images under head pose variations in our training data makes our method robust to this kind of constraint.

- $e \leq 0.05$: Corresponds to a high level of precision in estimation. It indicates a very low distances from the pupil center. Compared to [17] our method gives better results. The projection on Hough space implies an extension in the regression space of the forest. In addition, the absence of some typical examples like the presence of glasses in the training data in [17] paralyzes this method in some scenarios.

Figure 6 shows visual a illustration of 2D pupil estimation. The failures represented in the last row can be justified by the following:

- The failure of the face detector as shown in the first example of the last row which distorts the research area.

- The eyes appearance distorted by highlights on the glasses, dark illumination or eye closure.



(a) (b) (c)

Figure 7. (a) Comparison of the accuracy curve between our method and [17] for the talking video database. (b) The distribution of the normalized error over 1000 frames. (c) Some pupil estimations (the ground truth is represented by circles and the estimation by crosses), the first row representing some good estimations, the second row illustrating some failures of our method.

4.2. Quantitative results on videos

As in [17], we evaluate our method with the public database Talking Face video[2]. It contains 5000 frames of a person engaged in a conversation. A specific active appearance model [3] is trained to annotate the frames accurately. The forest trained in section 4.1 and the metric of equation 7 are used for the evaluation of our method.

We average the normalized error over all the frames. We obtain a mean error equal to 0.190. Because the authors of [17] not provide numerical results for their method, we tried to reproduce their accuracy curve at best and compare it to our approach. Figure 7a illustrates the comparison indicating that both methods give an estimation belonging to the pupil-radius ($e < 0.10$) over all the frames. The closure of eyes and the wrong annotations in some frames as shown in Figure 7c explain the peaks on the distribution of the normalized error.

4.3. The effect of forest parameters on the estimation

Our method is controlled by some parameters. The previous experiments were performed under optimal values of these parameters.

- **The number of trees** used for the estimation. Figure 8a illustrates the variation of the normalized error defined in equation 7 for 500 images from the BioID and talking video databases under different values of forest size. The error decreases by increasing the number of trees used for both databases (note the apparent gap between the two curves due to the different resolution of the images in the two databases). The normalized error is reduced by approximatively 30% compared to the initial value (from 0.055 to 0.040 for BioID and from 0.032 to 0.170 for the talking video) which is the result of output smoothing by the different trees. We noticed that, using more than 25 trees does not produce more precision, so we fix the optimal forest size to 25. Figure 8c shows the time in seconds needed to process the 500 frames under different sizes in the forest approach. The use of 25 trees gives an average fps of 30.

- **The number of patches** extracted from the testing image. Figure 8b represents the variation of the normalized error under different numbers of patches used for the estimation. The normalized error is reduced approximatively by 75% for the talking video database (from 0.082 to 0.02) and 45% for the BioID database (from 0.078 to 0.044). By increasing the number of patches, the trees get more information about the input test image which consequently gives more accurate estimations. In our experiments, according to the dimension of the image test (80×70), we noticed that 35 patches cover approximatively all the input information. Figure 8d shows the time needed to process 500 frames for different numbers of extracted patches.

- **The maximum variance** which is represented by the trace of the covariance of the offsets reaching each leaf is fixed to 800 and the probability $p(c/w)$ is fixed to 0.7. These values seems to provide good estimation results, a variance of 800 defines a voting area of approximatively (20×20) from the pupil center. The patch size of (16×16) gives an acceptable appearance which allows a good discrimination and generalization of the forest during the estimation.

Our real-time system is tested on a Intel Core i7 @ 2.70GHz with 8GB of RAM machine which is used for the training and the testing (the trees are trained sequentially). According to the low memory occupied by the trees and their quick response time for the regression, our results can be reproduced on a machine with lower processing performances. Figure (9) shows some successful qualitative estimations of the pupil location on still images with different scenarios, head pose variations, greater distances from the

[2] http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html

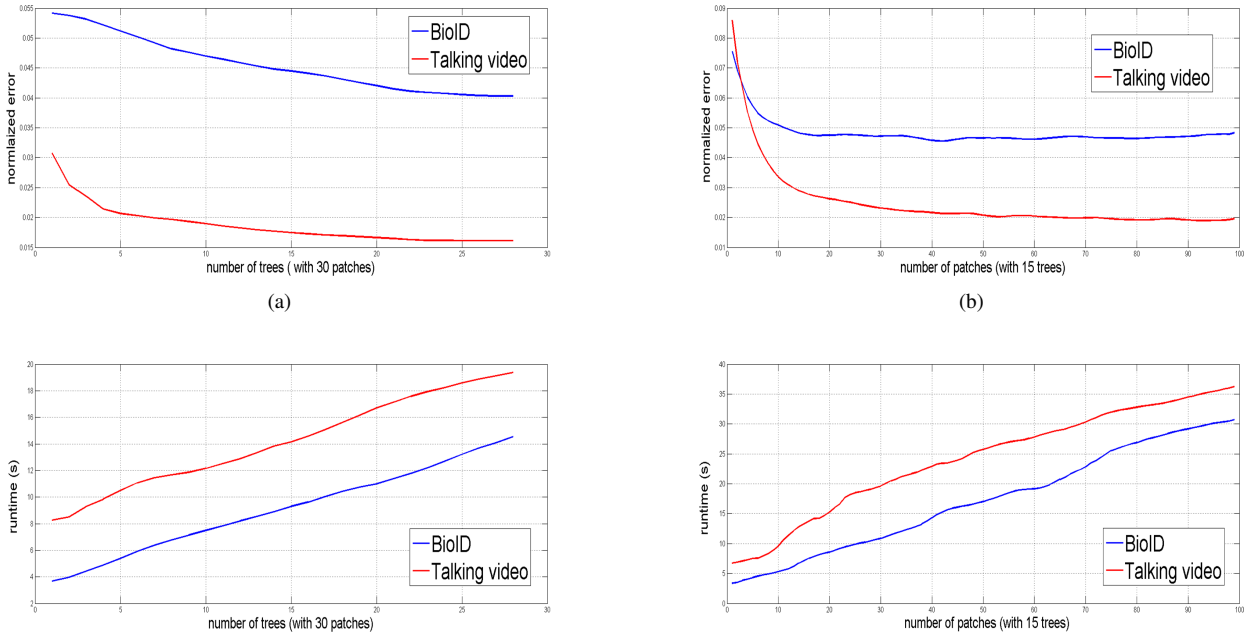


Figure 8. The forest parameters effect on the pupil location estimation on 500 frames from the BioID database labeled in blue and the talking video database labeled in red. a) The normalized error behavior under trees number variation with the number of patches extracted fixed to 30. b) The normalized error variation as a function of the number of patches extracted when the number of trees is fixed to 15. c) and d) represent the time needed to regress the output of all 500 frames under number of trees and the number of patches variations extracted respectively.

sensor, presence of glasses and multi-users tracking.

5. Conclusions

In this paper, we have proposed an approach for real-time pupil localization based on Hough regression forest. By exploiting the regression and classification information encoded by each tree, we construct a voting space which is the generalized Hough space. This space represents the response of each tree in the forest for each patch extracted from the test image which is extracted from a pyramid image. The maximum intensity is selected from this space after weighting all the results related to each image on the pyramid by their local maxima corresponding to 2D pupil location. By testing our approach on challenging public datasets, we demonstrate that our method yields a significant improvement in terms of robustness and precision compared to the state-of-the-art. These performances are directly linked to the generalization power of the trees built from the perturbations introduced in the training data and the extension of the regression ability by the generalized Hough space projection. The robustness of our approach can meet demanding eye-tracking requirements.

References

[1] V. Badrinarayanan, I. Budvytis, and R. Cipolla. Semi-supervised video segmentation using decision forests. In *Decision Forests for Computer Vision and Medical Image Analysis*, pages 229–244. Springer, 2013.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):532, 2001.

[3] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *TPAMI*, 2001.

[4] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu. Regression forests for efficient anatomy detection and localization in ct studies. In *Medical Computer Vision Workshop*. 2010.

[5] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, 2011.

[6] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *TPAMI*, 2011.

[7] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI*, 2001.

[8] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[9] N. Gourier, D. Hall, and J. L. Crowley. Estimating face orientation from robust detection of salient facial structures. In

Figure 9. Some successful qualitative results on still images of our approach. a) The first row shows some results from Pointing'04 database [9] with a strong head pose variations. b) The second row illustrates some results from Yale-extended database [7] with a strong illumination conditions variation. The third row shows some results on synthetic images constructed using a 3D morphable model [18]. The last row represents the results of the estimation in multi-users scenarios.

FG Net Workshop on Visual Observation of Deictic Gestures, 2004.

[10] K. R. Gray, P. Aljabar, R. A. Heckemann, A. Hammers, D. Rueckert, A. D. N. Initiative, et al. Random forest-based similarity measures for multi-modal classification of alzheimer's disease. *NeuroImage*, 65:167–175, 2013.

[11] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *TPAMI*, 2010.

[12] C. Huang, X. Ding, and C. Fang. Head pose estimation based on random forests for multiclass classification. In *ICPR*, 2010.

[13] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In *Audio-and video-based biometric person authentication*, pages 90–95, 2001.

[14] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR*, 2005.

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[16] R. Marée, L. Wehenkel, and P. Geurts. Extremely randomized trees and random subwindows for image classification, annotation, and retrieval. In *Decision Forests for Computer Vision and Medical Image Analysis*, pages 125–141. Springer, 2013.

[17] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer. Eye pupil localization with an ensemble of randomized trees. *Pattern recognition*, 47(2):578–587, 2014.

[18] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced Video and Signal Based Surveillance*, 2009.

[19] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[20] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[21] F. Timm and E. Barth. Accurate eye centre localisation by means of gradients. In *VISAPP*, 2011.

[22] R. Valenti and T. Gevers. Accurate eye center location and tracking using isophote curvature. In *CVPR*, 2008.

[23] A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza. Hybrid method based on topography for robust detection of iris center and eye corners. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(4):25, 2013.

[24] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[25] U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann. A comprehensive head pose and gaze database. In *Intelligent Environments, 2007. IE 07. 3rd IET International Conference on*, pages 455–458. IET, 2007.