Head Pose Free 3D Gaze Estimation Using RGB-D Camera

Amine Kacete, Renaud Séguier, Michel Collobert, Jérôme Royan

Head Pose Free 3D Gaze Estimation Using RGB-D Camera

Amine Kacete, Renaud Séguier, Michel Collobert, and Jérôme Royan
Institute of Research and Technology B-com

ABSTRACT

In this paper, we propose an approach for 3D gaze estimation under head pose variation using RGB-D camera. Our method uses a 3D eye model to determine the 3D optical axis and infer the 3D visual axis. For this, we estimate robustly user head pose parameters and eye pupil locations with an ensembles of randomized trees trained with an important annotated training sets. After projecting eye pupil locations in the sensor coordinate system using the sensor intrinsic parameters and a one-time simple calibration by gazing a known 3D target under different directions, the 3D eyeball centers are determined for a specific user for both eyes yielding the determination of the visual axis. Experimental results demonstrate that our method shows a good gaze estimation accuracy even if the environment is highly unconstrained namely large user-sensor distances (> 1m50) unlike state-of-the-art methods which deal with relatively small distances (< 1m).

Keywords: Gaze estimation, pupil localization, head pose estimation, 3D eyeball, Random Forest, RGB-D camera.

1. INTRODUCTION

The point-of-gaze (POG) is the intersection of the two visual axes of the eyes. Many applications use this information, typically, in human computer interaction (HCI), driver's behavior analysis, human cognitive state determination and monitoring security. Recent gaze estimation methods can be divided in two global categories: appearance-based and feature-based methods.

Appearance-based methods learn a mapping function f ($y = f(x)$) where x represents the input usually defined as the eye image appearances, and y is the gaze information output. Many algorithms have been proposed, [1] trained a neural network with $2k$ samples to learn the mapping function. [2] collected 252 training samples to build a manifold of the local linearity related to the eye appearances and estimated an unknown sample using a linear interpolation. [3] trained a semi-supervised Gaussian Process on 80 samples relatively sparse. [4] proposed a Support Vector Regressor to achieve a highly non-linear mapping. However, the extracted eye images appearances exploited by these methods are very variable with head pose changes. [5] used a specific head mounted hardware to track gaze in unconstrained environments. [6] used an incremental learning and built a specific cluster for each head pose. [7] used l-optimization to adapt the gaze manifold. [8] learned a person-specific 3D model which is used to estimate head pose and then normalize the eye images to a frontal view for the learning.

Figure 1. Gaze estimation by our approach. For each eye, a gaze vector is estimated (blue lines), the ground truth is represented by the green lines which connect each eye with a tracked target in 3D.

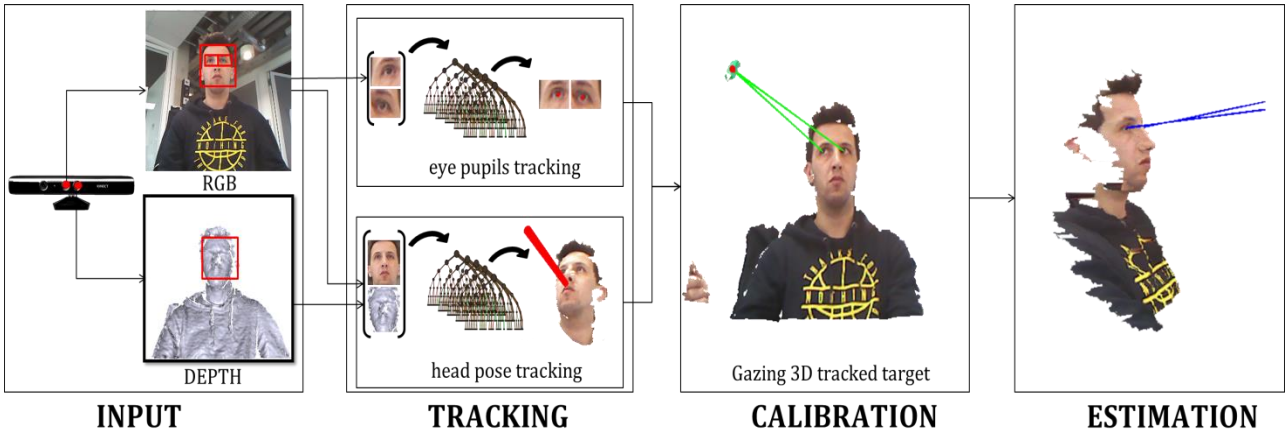

Figure 2. Overview of our approach. Four principal blocks can be distinguished. Input block describes the data grabbed from the depth sensor used in our method. Tracking block illustrates two global components, head pose and eye pupils estimation respectively using RGB-D cues. Using the computed information from the previous blocks, Calibration fixes for a specific user some parameters related to the eye geometry (performed by gazing an known target in 3D). Finally, estimation block gives gaze vectors for each eye.

Feature-based methods rely on the extraction of some features such as pupil centers, eyes corners, iris contour or corneal reflection which are used to build a 3D eye model and determine the visual axis. [9] and [10] used the pupil center corneal reflection extracted from IR lights which are used to illuminate the eye regions. [11] and [12] estimated iris shape by fitting an ellipse to infer the gaze. [13] and [14] estimated the gaze direction from the 2D pupils and corners locations in the eye image. All the above methods simplify the anatomical structure of the eyeball and define the gaze direction as the optical axis. [15] proposed an extended 3D eye model based on the pupil and the corners and estimate the visual axis but still requiring a high image resolution to detect the corners accurately, in addition, they manually labeled pupils centers.

The main challenge of the feature-based methods consists in localizing user's eye key points (pupil and corners) and head with high accuracy which are directly involved in the final estimation. In this paper, we propose an approach based on robust head pose estimation and accurate pupil localization based on random regression forest with one-time specific-person calibration to build the 3D eyeball model and estimate visual axis using RGB-D camera. Fig. 1. shows an example of gaze estimation performed by our method.

The following sections are organized as follows: In Sec.2, we present our gaze estimation algorithm describing head pose and pupils locations estimation, then we discuss the eyeball parameters calibration. In Sec.3, we show the results of our experiments under different scenarios. Sec.4 concludes our work.

2. GAZE ESTIMATION ALGORITHM

Fig. 2. shows an overview of our gaze estimation system. Using the RGB-D information grabbed from Kinect sensor as input, we estimate both head pose parameters (expressed in the Kinect coordinate system) and the 2D eye pupil locations projected in 3D using the sensor intrinsic parameters. After gazing a known target tracked in 3D, we drive a calibration equation to calculate the eyeball centers. Finally, knowing eyeball centers, the cornea centers can be determined yielding the determination of the optical axis as the connecting line of eyeball and cornea centers. Using a predefined relation between optical and visual axis, we estimate the final gaze vector for each eye. Each part of our pipeline is discussed and detailed as follows:

2.1 Input

We grabbed the RGB and depth map at (1280-960) and (320-240) resolutions respectively at 15 fps (the fps is constrained by the RGB sensor resolution of the Kinect sensor). Using the known Kinect intrinsic parameters and a predefined rigid transformation between the RGB and depth sensors, each depth value can be projected in 3D using the pinhole model as follows:

$$\begin{cases} x_d = \dfrac{d(u_d - c_x)}{f_x} \\ y_d = \dfrac{d(v_d - c_y)}{f_y} \\ z_d = d \end{cases} \quad (1)$$

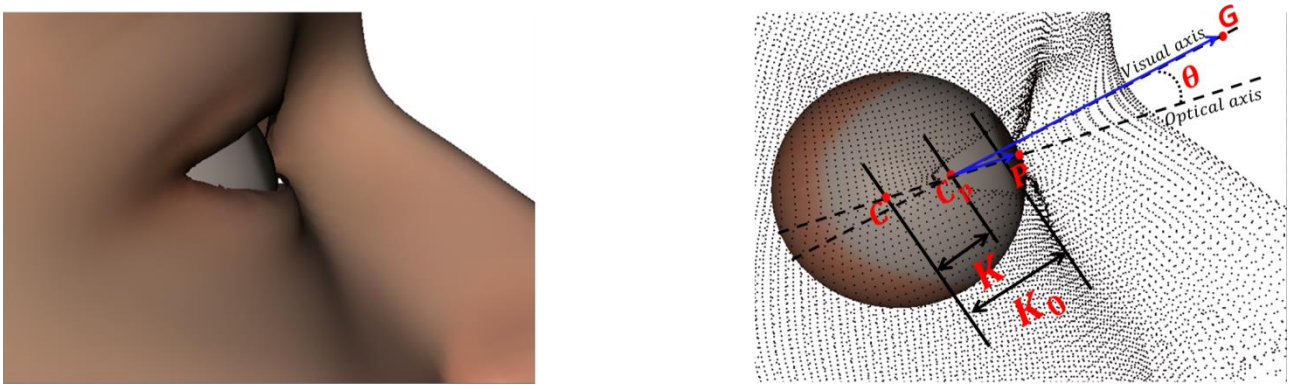

Figure 3. 3D eyeball model. C, C_p and P represent eye, cornea and pupil centers respectively. For human eye, [CC_p] and [CP] distances are constants. Visual and optical axes are represented in blue connecting C_p-G and C_p -G respectively. The dotted bow represents the angular relation between optical and visual axis (only vertical angle θ is illustrated here)

Where d represents a depth value with its coordinates (u_d,v_d) and sensor intrinsic parameters (f_x,f_y,c_x,c_y). (x_d,y_d,z_d) represent the final 3D projections. To produce a textured mesh as illustrated in Fig. 2., a rigid mapping from RGB and depth sensors has to be established.

2.2 Tracking

According to the high accuracy in estimating head pose and eye pupil locations needed to produce a sufficiently accurate gaze estimation in our method, we decided to use Random Forest algorithm to handle these tasks in hand.

Introduced by [16], Random Forest is a set of weak tree predictors which splits the initial problem into two low complex problems in a recursive way. At each node, a simple binary test is performed, according to the result of the test, a data sample is directed towards the left or the right child. The tests are selected to achieve an optimal clustering. The terminal nodes of the tree called leaves, store the estimation models approximating the best the desired output. This technique is widely used in computer vision problems such as classifications: [17] [18] and regression: [19] [20] [21]. We trained for each component a forest $\mathcal{T} = \{\chi_t\}$ in a supervised way using a set of annotated patches $\{\mathcal{P}_i = (\mathcal{J}_i^c, g_i)\}$ where $\mathcal{J}_i^c$ represents the appearance of the i^{th} patch composed of c channel used for the training and g_i is the output. Supervising each tree χ_t consists in finding at each non-leaf node the optimal binary test t^* that minimizes the node impurity. Minimizing the node impurity is achieved by maximizing the information gain defined as the differential entropy of the set of patches at parent node $\mathcal{P}$ minus the weighted sum of the differential entropies computed at the children $\mathcal{P}_\mathcal{L}$ and $\mathcal{P}_\mathcal{R}$ defined as:

$$E = H(\mathcal{P}) - (\omega_\mathcal{L} H(\mathcal{P}_\mathcal{L}) - \omega_\mathcal{R} H(\mathcal{P}_\mathcal{R})) \tag{2}$$

The weights $\omega_{j\in\{\mathcal{R},\mathcal{L}\}}$ are defined as the ratio of patches reached to the parent and the right or left child respectively, *i.e.*, $|\mathcal{P}_{j\in\{\mathcal{R},\mathcal{L}\}}| / |\mathcal{P}|$. Assuming that the output g at each node is a random variable with a multivariate Gaussian distribution such as $p(g) = \mathcal{N}(g, \bar{g}, \Sigma)$, it allows us to rewrite equation (2) as follows:

$$E = \log|\Sigma(\mathcal{P})| - \sum_{j\in\{\mathcal{L},\mathcal{R}\}} \omega_j \log|\Sigma_j(\mathcal{P}_j)| \tag{3}$$

The learning process finishes when the data reach a predefined maximum depth value of the tree or the number of patches let down a threshold value yielding the creation of the leaves. A leaf l stores the mean of all the gaze vectors which reached it with the corresponding covariance.

Head pose we trained our head pose forest $\mathcal{T}_{\text{head}}$ as done in [22] on a large synthetic data training set rendered using a 3D morphable model. $\mathcal{J}$ is composed of two channels, depth and gray scale information cropped around the face (after performing a face detection step). The output g encodes head gravity center (x_{hg},y_{hg},z_{hg}) (as a translation matrix T) and the Euler's rotation angles (pitch, roll and yaw) converted to a rotation matrix R. According to the sensor coordinate system, a global rigid transformation $\mathcal{O}$ can be formulated as follows:

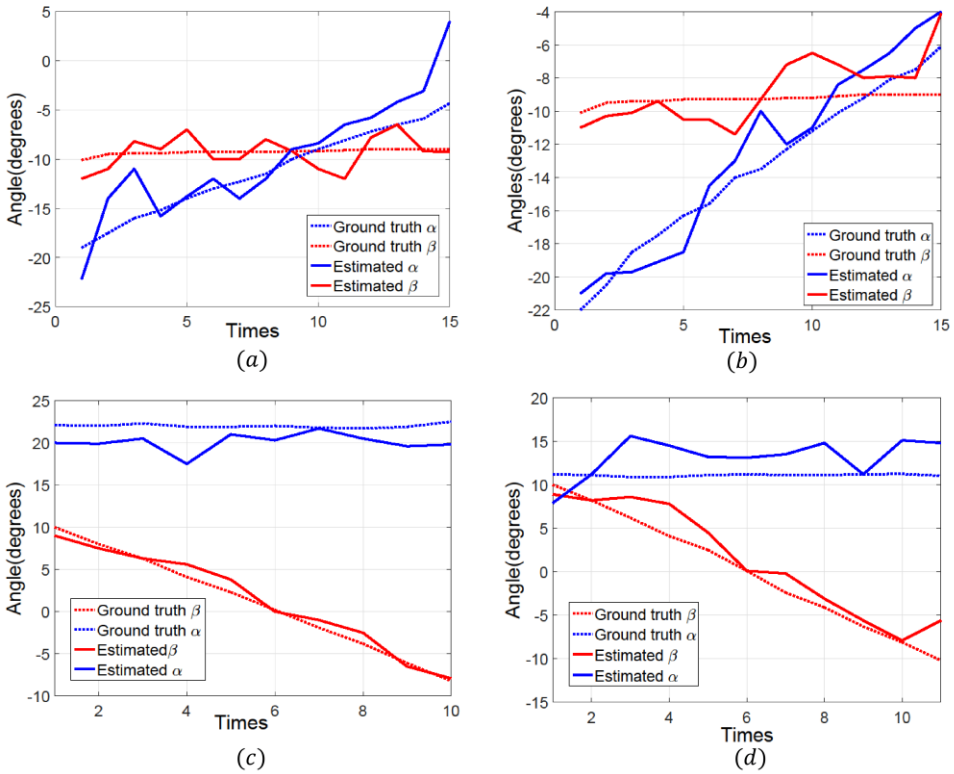

Figure 4. Gaze estimation error at 75 cm. (a) and (c) describe gaze estimation errors for upward and rightward moving of the target for right eye. (b) and (d) illustrate errors for left eye under the same scenario.

$$\mathcal{O} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{4}$$

The part tracking in Fig. 2 describes head pose component, red cylinder illustrates the final estimation. A total of 20 trees are trained on 1M training data.

Eye pupils we trained our eye pupils forest $\mathcal{T}_{\text{pupils}}$ on a large real data training set using the public datasets [23] and [24]. Patch appearance $\mathcal{J}$ is defined with one channel as the gray scale values around eye regions cropped from the face. In this case, the output g represents the 2D pupils (u_{p},v_{p}) in the 2D image coordinate system. Using equation (1), we calculate the 3D locations P(x_{P},y_{P},z_{P}). Fig. 2 illustrates eye pupils localization in the tracking part. A total of 30 trees are trained on 500k training data.

2.3 Calibration

To compute the eyeball center C, we assume a known target gaze point G(x_{G},y_{G},z_{G}) as illustrated in the calibration part of Fig. 2. When the user is focusing at G, the angle between the optical axis $\overrightarrow{C_P P}$ and the visual axis $\overrightarrow{C_P G}$ would be θ which is a constant value. [7] describes an additional relation between the two axis as follows:

$$\frac{\overrightarrow{C_p G} . \overrightarrow{C_p P}}{\left\|\overrightarrow{C_p G}\right\| \left\|\overrightarrow{C_p P}\right\|} = \cos(\theta) \tag{5}$$

As the distances K_0 and K are constant, a relation between C and C_{P} (Fig. 3 illustrates the existing relationships between these points) can be established as follows:

$$C_p = C + \frac{K_0}{K}(P - C) \tag{6}$$

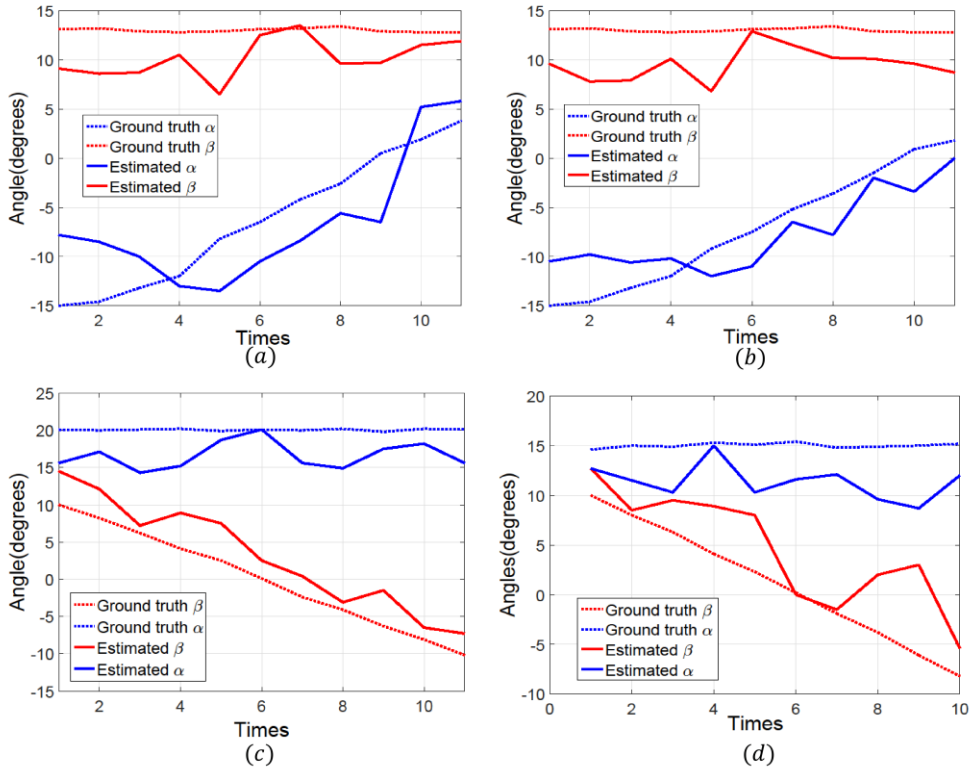

Figure 5. Gaze estimation error at 150 cm. (a) and (c) describe gaze estimation errors for upward and rightward moving of the target for right eye. (b) and (d) illustrate errors for left eye under the same scenario.

Using Levenberg-Marquardt optimization, the non-linear equation (2) can be solved. By using the equation (3), the eyeball center can be initialized at C_0 and transformed to the Kinect coordinate system as follows:

$$C = R * C_0 + T \tag{7}$$

2.4 Estimation

Knowing eyeball center C and the pupil P at each frame, cornea center C_P can be calculated. Thus, the optical axis can be estimated, by adding the constant angles values, the visual axis can calculated and the gaze vectors can be expressed as vertical and horizontal angles (α,β) for each eye.

3. EXPERIMENT RESULT

In our experiments some parameters related to eyeball geometry are fixed beforehand. The constants K_0 and K inside the eyeball are fixed as the average human values to 5.3 cm and 13.1 cm respectively. The horizontal and vertical angles between visual axis and optical axis are fixed to 5° and 1.5° respectively as done in [9]. We calibrate the eyeball for a specific user by solving the non-linear equation (5) with 5 gaze samples recorded under different directions.

To evaluate our method, we design a target point represented by a green marker cap which can be easily tracked in 3D (based on color segmentation as done for the calibration step) moving in front of the user. We tested gaze estimation accuracy when the target is moving upward and rightward with two user-sensor distances (75 cm and 150 cm). Fig. 4. shows the comparison diagram between ground truth and our estimation, for the upward scenario, only α is changing while the β is changing for the rightward one for both eyes. As we can see, our estimation is close to ground truth, comparing to [8], our method gives better results and the average error remains below 5.5°. For 150 cm distance, RGB and depth image resolutions decrease significantly giving a less accurate head and pupils tracking producing higher gaze estimation errors. Fig. 5. shows the gap between estimation and ground truth, however errors still acceptable (less than 7.5°). Despite robustness of our tracking component, the difference in RGB and depth resolution (which is a hardware

limitation) makes projection of the 2D pupil locations in the sensor coordinate system very sensitive giving sometimes instable gaze vectors.

Tab. I. quantifies the gaze estimation errors for both 75 cm and 150 cm scenarios in upward and rightward configurations.

Table I. Gaze estimation error under two user-sensor distances, 75 cm and 150 cm respectively.

Directions	75 cm				150 cm			
	Error right eye		Error left eye		Error right eye		Error left eye	
	α	β	α	β	α	β	α	B
upward	3.81°	2.95°	3.12°	5.23°	4.12°	3.56°	5.12°	7.51°
rightward	4.13°	3.11°	5.27°	3.20°	5.20°	3.75°	6.11°	4.58°

4. CONCLUSION

In this paper, we described an algorithm for 3D gaze estimation based on robust head pose estimation and eye pupils localization using an ensemble of tree predictors learned on important annotated training data. To set up the 3D eye model, we fixed some constants relative to eye geometry to average human values and calibrated eyeball centers which allowed us to determine the visual axis and the point of regard as the intersection of the two axes for both eyes. Our experiments showed that our method can achieve good accuracy and handle important head movements and large user-sensor distances.

REFERENCES

[1] S. Baluja and D. Pomerleau, "Non-intrusive gaze tracking using artificial neural networks, " 1994.

[2] K.-H. Tan, D. J. Kriegman and N. Ahuja, "Appearance-based eye gaze estimation," in *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, 2002.

[3] O. Williams, A. Blake and R. Cipolla, "Sparse and Semi-supervised Visual Mapping with the S3GP," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006.

[4] Z. Zhu, Q. Ji and K. P. Bennett, "Nonlinear eye gaze mapping function estimation via support vector regression," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006.

[5] B. Noris, J.-B. Keller and A. Billard, "A wearable gaze tracking system for children in unconstrained environments," *Computer Vision and Image Understanding*, vol. 115, no. 4, pp. 476-486, 2011.

[6] Y. Sugano, Y. Matsushita, Y. Sato and H. Koike, "An incremental learning method for unconstrained gaze estimation," in *Computer Vision--ECCV 2008*, Springer, 2008, pp. 656-667.

[7] F. Lu, Y. Sugano, T. Okabe and Y. Sato, "Inferring human gaze from appearance via adaptive linear regression," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011.

[8] K. A. F. Mora and J.-M. Odobez, "Gaze estimation from multimodal kinect data," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference*

[9] E. D. Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *Biomedical Engineering, IEEE Transactions on*, vol. 53, no. 6, pp. 1124-1133, 2006.

[10] Z. Zhu and Q. Ji, "Novel eye gaze tracking techniques under natural head movement," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 12, pp. 2246-2260, 2007.

[11] J.-G. Wang and E. Sung, "Study on eye gaze estimation," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 32, no. 3, pp. 332-350, 2002.

[12] S. Kohlbecher, S. Bardinst, K. Bartl, E. Schneider, T. Poitschke and M. Ablassmeier, "Calibration-free eye tracking by reconstruction of the pupil ellipse in 3D space," in *Proceedings of the 2008 symposium on Eye*

tracking research \& applications, 2008.

[13] T. Ishikawa, "Passive driver gaze tracking with active appearance models," 2004.

[14] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000

[15] J. Chen and Q. Ji, "3D gaze estimation with a single camera without IR illumination," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008.

[16] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, p. 5–32, 2001.

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

[18] 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*, Springer, 2013, pp. 125-141.

[19] A. Criminisi, J. Shotton, D. Robertson et E. Konukoglu, «Regression forests for efficient anatomy detection and localization in CT studies,» chez *Medical Computer Vision Workshop*, 2010.

[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*, vol. 56, no. 1, pp. 116-124, 2013.

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

[22] G. Fanelli, J. Gall and L. Van Gool, "Real time head pose estimation with random regression forests," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011.

[23] 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*, 2007.

[24] 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)*, vol. 9, no. 4, p. 25, 2013.

AUTHORS' BACKGROUND

Your Name	Title	Research Field	Personal website
Amine Kacete	PhD candidate	Computer vision	http://www.rennes.supelec.fr/ren/perso/kacete_ami/
Renaud Séguier	Full professor	Signal processing	http://www.rennes.supelec.fr/ren/perso/rseguier/
Michel Collobert	Senior engineer	Machine learning	http://mcollo.pagesperso-orange.fr/
Jérôme Royan	PhD-Engineer	Augmented reality	https://b-com.com/fr/institut/galaxie-bcom/jérôme-royan