Can Synthetic Data Handle Unconstrained Gaze Estimation ?

A. Kacete[1] R. Séguier[2] M. Collobert[3] J.Royan[1]

[1] Institute of Research and Technology b-com
[2] Centralesupelec
[3] Orange Labs

amine.kacete@b-com.com

Abstract

In this article, we aim at solving unconstrained gaze estimation problem using appearance-based approach. Unlike previous methods working in relatively constrained environment, we propose an approach that allows free head motion and significant user-sensor distances using RGB-D sensor. Our paper presents the following contributions : (i) A direct estimation by inferring gaze information from RGB eyes and depth face appearances ;(ii) A channel selection strategy during the learning to evaluate the involvement of each channel in the final prediction ; (iii) Adapting a 3D face morphable model by integrating a parametric gaze model to render an important synthetic RGB-D training set. We also collect real labeled samples using Kinect sensor that allows for evaluating the potential of synthetic learning in handling real configurations and establish an objective comparison with real learning. Results on several users demonstrate the great potential of our approach.

Keywords

Gaze estimation, Eye tracking, Random Forest, Synthetic data, 3D morphable model.

1 Introduction

Gaze estimation plays a key role in several computer vision applications. In facial expression recognition fields, it allows access to important information such as the cognitive and expressive state of the person. In human behavior analysis, it allows the point of interest of the user, which represents the input of various devices such as user attention while driving and helping disabled people. Several industrial solutions are commercialized. They provide good accuracy on gaze estimation. 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, but they remain very sensitive to illumination conditions. Recently, researches focus on using low cost devices such as monocular cameras, a comprehensive survey presented in [9], considers two main categories of gaze estimation, features-based methods and appearance-based methods.

1.1 Feature-based methods

These methods rely on the extraction of some features such as the pupil center, the eye corners, the iris contour or the corneal reflection, which are used to build a 3D eye model and determine the visual axis. [8] and [25] used the pupil center corneal reflection, from the IR lights which are used to illuminate the eye regions from different directions giving different image appearances, the corneal reflection is built by subtracting these images. [22] and [11] estimated the shape of the iris by fitting an ellipse to infer the gaze. [15] and [10] estimate the gaze direction from the 2D locations of the pupils and the corners in the eye image. All the above methods simplify the anatomical structure of the eyeball and define the gaze direction as the optical axis. [4] proposed an extended 3D eye model based on the pupil and the corners to estimate the visual axis but still require a high image resolution to detect the corners accurately, in addition, they manually labeled the pupils centers. The main limitation of these methods lies in the direct link between their gaze estimation precision and the accuracy of the eye's key-points localization (pupil, corners etc.) which requires a high image resolution and small head pose changes.

1.2 Appearance-based method

Appearance-based methods learn the mapping from the eye image appearances space to the gaze estimation space. Many algorithms have been proposed. [1] trained a neural network with 2k samples to learn the mapping function. [20] proposed a weighted linear interpolation to estimate an unknown gaze point from 252 sparse samples. [23] trained a semi-supervised Gaussian process on 80 samples relatively sparse. [26] proposed support vector repressors to achieve a high non-linear mapping . [19] proposed an incremental learning strategy using an on-line sample acquisition from a video stream updating the mapping function for a number of limited head pose configurations. [14] introduced the adaptive linear regression for the learning on a very sparse training set. The accuracy of these previous approaches is significantly affected in unconstrai-

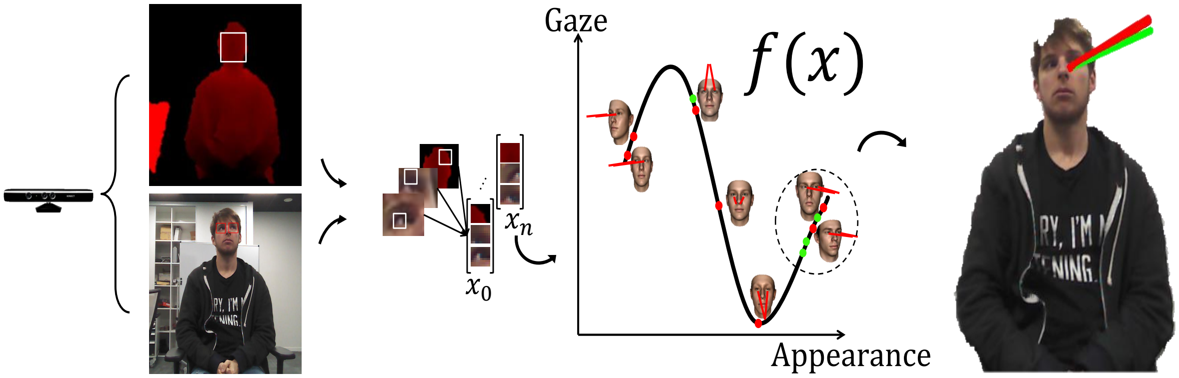


FIGURE 1 – Automatic gaze estimation based on our approach. We build a 3-channels global vector represented by the two RGB eye images and the face depth information using the depth sensor multimodal data, we extract a set of patches and project it through the forest represented here as the mapping function $f(x)$ (the learned gaze sample clusters are defined as the red centroid points). Each single tree casts votes for each patch (defined as the green points). By performing a non-parametric clustering technique, a final estimation is calculated (represented as the green line, the red one defines the ground truth).

ned environment with high head pose changes. Recently, some methods aim to manage such trouble by considering gaze estimation and head pose as two independent geometrical components, these approaches can be seen as semi appearance-based methods. [13] proposed to separate head pose component from the global gaze estimation system. By performing an initial estimation under frontal configuration assumption and a geometric compensation with the head pose parameters, the final gaze estimation is inferred. Using the same paradigm, [16] projected the training gaze sample in frontal manifold using a frontalization step based on the head pose parameters calculated using a specific 3D model fitting. These last two methods solved the problem of head changes successfully but still working under low user-camera distances. To cover all the eye image appearance variability [24] recorded around 200k training samples and used a deeper strategy using a convolutional neuronal network to learn a very robust mapping function achieving a high gaze estimation accuracy but very constrained by an important computational time making this method not real-time.

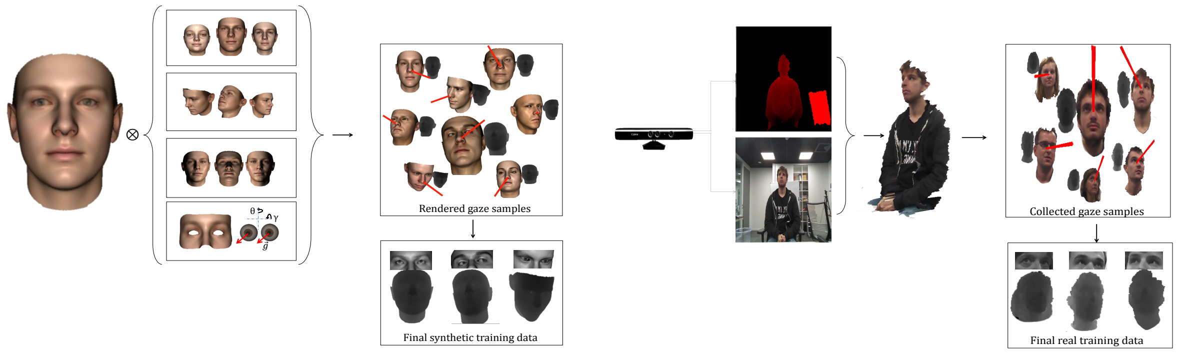
In this paper, we consider the high non-linear problem of gaze estimation under head pose changes and large user-sensor distances as a regression task. To learn such mapping robustly, we propose a novel approach that considers a global gaze manifold instead of learning in frontal configurations and geometrically correct the final estimation using head pose parameters as usually done. We train an ensemble of regression trees able to capture robustly gaze information on an important 3-channels training samples (channel$^{(0)}$,channel$^{(1)}$ defines the gray scale images relative to right and left eye respectively, channel$^{(2)}$ defines the depth image of the face) organized as a set of patches (where a patch defines a small group of nearby pixels). We apply a channel-selection during the training to evaluate the importance and involvement of each channel in the final estimation. We define the gaze vector g as the vector stretching the gravity center of the face and the gazed 3D point. To provide a significant set of training data for learning the trees, we render a very important amount of gaze samples using a 3D statistical morphable model with integrating dynamic gaze model. We also build an important gaze database recorded with the Microsoft Kinect sensor. Rendered synthetic data are exclusively used for the learning and real data are used for both learning and testing.
Fig. 1 describes an overview of our automatic gaze estimation system. The rest of the paper is organized as follows : Sec. 2 describes our method in details. In Sec 3, we detail how training data are generated. Sec. 4 describes the experiments and evaluates the precision of our approach. Sec. 5 concludes the paper.

2 Our method

We use randomized regression trees to estimate the two angles (θ, γ), which represent the horizontal and vertical orientation of the gaze vector g, from the RGB and depth cues combined on 3-channels patches. In Sec. 2.1, we provide some background on regression trees. In Sec. 2.2 and Sec. 2.3 we detail the training and testing steps of our forests respectively. Sec.3 describes the generation of our training RGB-D gaze samples.
For the next sections, we define also head pose parameters as $\mathcal{H}$ (with $\mathcal{H} : [R|T]$).

2.1 Random regression forest

Recently, many applications in computer vision have used Random Forest to achieve the mapping from complex input spaces into discrete or continuous output spaces. Introduced by [2], randomized trees deal with different tasks such as classification in [7, 12] and regression in [18, 5].



(a)

(b)

FIGURE 2 – Data generation. (a) represents synthetic data rendering using the 3D morphable model of [17]. By introducing some variabilities such as identity, head pose changes and lighting conditions, we integrated a dynamic gaze model (represented by two global textured spheres), we rendered the final RGB-D gaze training samples with the correspondent gaze annotation illustrated in red line. (b) We performed the same strategy using real data grabbed from the multimodal Kinect sensor by introducing the same previous variabilities. To obtain gaze annotation, a 2D moving point is gazed by the user (knowing the rigid transformation sensor-screen, the stretching vector from user head gravity and the projection of the moving point in the word coordinate can be calculated). These real data are principally used to evaluate accurately the performance of the synthetic data in handling gaze estimation.

Regression forest is an ensemble of trees predictors which splits the initial problem in 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. To achieve high generalization, the trees are trained in a decorrelated way (with introducing randomness in both the training data provided for each tree and the set of binary tests).

2.2 Training

We supervised the training of each tree T in the forest $\mathcal{T} = \{T_t\}$ using a set of annotated patches $\{\mathcal{P}_i = (\mathcal{I}_i^c, g_i)\}$ randomly selected from the training data where :

- $\mathcal{I}_i^c$ represents the extracted visual features vector from a given patch $\mathcal{P}_i$, c defines the feature channel. We used 3 channels namely the two grayscale intensities extracted from the two eye images and the depth values extracted from the face.
- g_i represents the output gaze vector represented with two components (θ, γ).

Starting from the root, at each non-leaf node, we define a simple binary test $t_{x_1,y_1,x_2,y_2,c,\tau}$:

$$\begin{cases} 1, & if \;\; \mathcal{I}_i^c(x_1,y_1) - \mathcal{I}_i^c(x_2,y_2) \leq \tau \\ 0, & otherwise \end{cases}$$

where $(\mathcal{I}_i^c(x_1,y_1) - \mathcal{I}_i^c(x_2,y_2))$ represents the difference of intensity between two locations (x_1,y_1) and (x_2,y_2) in the channel c. Supervising the training consists in finding at each non-leaf node the optimal binary test t^* that maximizes the purity of the data clustering. Maximizing the clustering purity is achieved by maximizing the information gain defined as the differential entropy H 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 $\mathcal{P}_\mathcal{R}$:

$$E = H(\mathcal{P}) - (\omega_\mathcal{L} H(\mathcal{P}_\mathcal{L}) + \omega_\mathcal{R} H(\mathcal{P}_\mathcal{R})) \qquad (1)$$

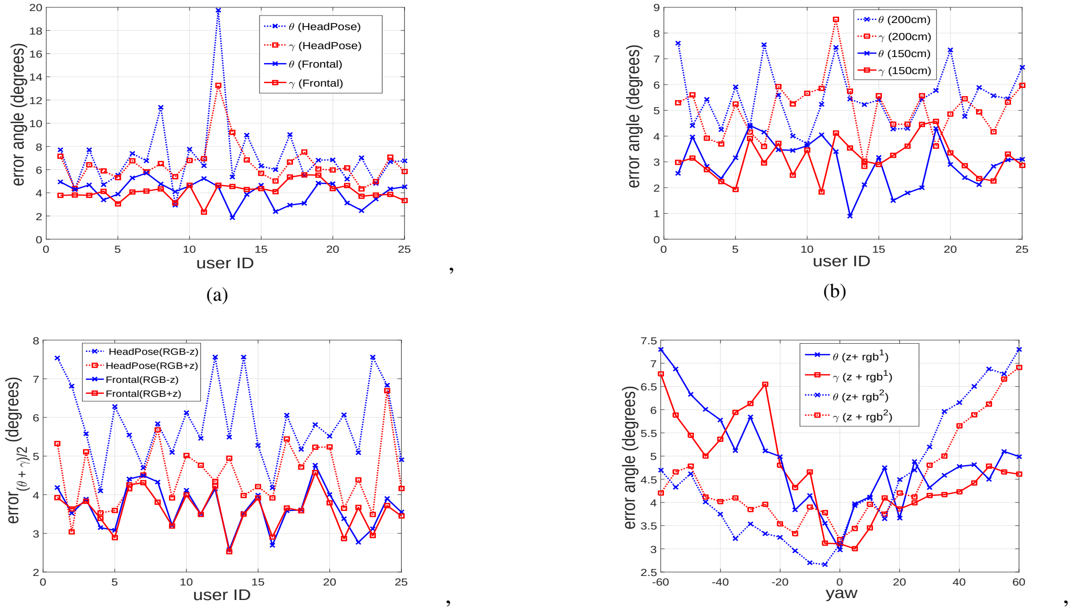
The weights $\omega_{j\in\{\mathcal{R},\mathcal{L}\}}$ are defined as the ratio of patches reached to the parent and the right or left child respectively, $i.e.$, $\frac{|\mathcal{P}_{j\in\{\mathcal{R},\mathcal{L}\}}|}{|\mathcal{P}|}$. Assuming that the gaze vector g at each node is a random variable with a multivariate Gaussian distribution such as $p(g) = \mathcal{N}(g, \bar{g}, \Sigma)$, allows us to rewrite Eq. 1 as follows :

$$E = \log|\Sigma(\mathcal{P})| - (\omega_\mathcal{L} \log|\Sigma(\mathcal{P}_\mathcal{L})| + \omega_\mathcal{R} \log|\Sigma(\mathcal{P}_\mathcal{R})|)| \qquad (2)$$

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

2.3 Testing

Given an unseen instance, we extract a set of patches from the RGB eye regions and the face depth information after a face detection step. Each patch is passed through all the learned trees in the forest. Using the optimal stored binary test, each tree processes the patch until reaching a leaf. The gaze vector estimation according to a single tree is given by the reached leaf l in terms of the stored distribution



(a) ,

(b)

(c) ,

(d) ,

FIGURE 3 – (a) the mean error for the two gaze directions under frontal and head pose changes. (b) the mean error for the two gaze directions under two distances from the sensor. (c) mean error over the two directions with and without using channel$^{(2)}$ in frontal and head pose configurations respectively. (d) mean error of the two direction over head pose variation (yaw angle variation) with different channels combination

$p(g|l) = \mathcal{N}(g, \bar{g}, \Sigma)$. The gaze vector estimation for a given patch $\mathcal{P}_i$ over all the trees is calculated as follows :

$$p(g|\mathcal{P}_i) = \frac{1}{|\mathcal{T}|}\sum_t p\big(g|l_t(\mathcal{P}_i)\big) \qquad (3)$$

All the estimations corresponding to the extracted patches are grouped in a set of votes. Before performing the clustering of these votes, we discard the estimations from the leaves with high variance considered as non-informative. To locate the centroid of the cluster of the votes, we perform 5 mean-shift iterations using a Gaussian kernel.

3 Data generation

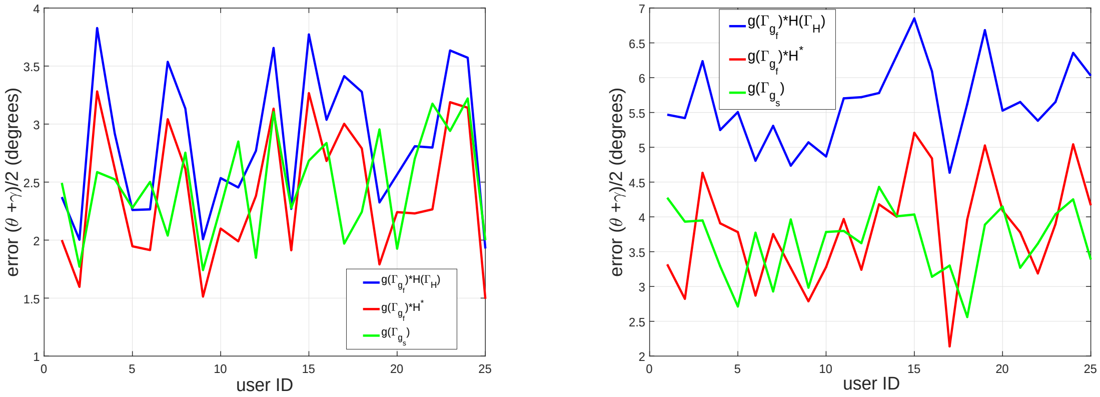
To provide a representative training dataset, we use two types of data : synthetic and real data.

3.1 Synthetic data

In Computer vision community, machine learning techniques are considered as a very elegant way to tackle problems. They demonstrated a great potential in terms of efficiency and robustness. Nevertheless to achieve a high generalization across unseen scenarios, these methods often require a very representative training data set. Given that the building of high amount of labeled data is a very tedious process, synthetic data represent a promising solution. Indeed, the annotation is performed automatically instead of manual labeling. [3] developed an iterative model based on Gabor-filters applied on an empty image containing some seed points to render a fingerprint training samples. [27] rendered iris image samples obtained from a 2D polar projection of a cylindrical representation of continuous fibers. [21] improved face authentication by generating multiple virtual images using simple geometric transformations. [18] used a motion capture strategy to record RGB and depth cues of the body part movements, by varying body size and shape, scene position, camera position and mirroring the recorded data, they synthesize a highly varied training allowing a robust body part pose estimation. [6] tackled the head pose estimation problem with synthetic depth images by rendering an enormous amount of training data using a 3D statistical morphable model.
In our method, we first generate our synthetic training gaze samples by rendering the 3D morphable model proposed by [17]. This model is built from around 200 scans of human faces. It contains a very high mesh density including the face, frontal neck and the ears. The shape and texture of the model is composed as a linear combination of 199 components. They can be deformed according to the following equation :

$$\mathcal{A} = \mathcal{A}_0 + \mathcal{M}_{\mathcal{A}}\alpha \qquad (4)$$

where $\mathcal{A}$ can denote the generated texture or shape respectively. $\mathcal{A}_0$ denotes the mean, $\mathcal{M}$ represents the basis com-



(a)

(b)

FIGURE 4 – Comparison of our approach to semi appearance-based approach using exclusively synthetic data. Our approach (in red) performs gaze estimation using RGB-D cues assuming a global gaze manifold(under head pose changes) in red. Semi appearance-based approach (in green) performs gaze estimation assuming frontal configuration with a geometrical correction using estimated head pose parameters. In blue : a semi appearance-based approach with ideal head pose parameters. (a) and (b) : mean gaze error across the two directions for frontal and head pose configuration respectively.

ponents perturbed with parameters α.

Fig.2a shows an overall of the generation process. To introduce face identity variation, we perturb the first 50 basis components of the shape and texture by ± 1.5 of the standard deviation of each mode. To render images in different head pose configurations, we apply random rigid transformations on the model : the rotations spans ± 60 for yaw and ± 40 for pitch and we translate the model along the z axis within 200 cm range for scale variability. Furthermore, to produce illumination variability, we generate light sources with different intensities and directions. Unfortunately, the basis components related to the shape and the texture of this model do not monitor the gaze direction.

So, to integrate a dynamic gaze system to the model able to generate different gaze direction instances, we decided to delete all the vertices related to the eye regions. Two spheres are placed as the eyeballs instead. We fix the diameters to human average eyeball namely 25 mm. We use different textures for the eyeballs to handle iris appearance variability. Moreover to control eyelids movements resulting from the gazing up and down, we introduce a linear translation for each vertex surrounding the eye regions as blendshapes. By defining the starting and the ending position in the global mesh, all the coefficients of the linear translations can be calculated. Thanks to the topology of the model, all these modifications keep the same behavior under identity variation. To generate gaze samples, we apply random rotations to the eyeballs, the gaze information angles can be easily computed knowing the location of the eyeballs.

3.2 Real data

In the other hand, we recorded real gaze sample data using Microsoft Kinect sensor. The database contains 17k RGB-D images of 42 people (15 females and 27 males, 4 with glasses and 38 without glasses) gazing different targets displayed on a screen. The subject performed 4 scenarios, gazing with a fixed head at roughly $d_0 = 150$ cm from the sensor, gazing at same distance d_0 under head pose changes and the two others scenarios are performed at about $d_1 = 200$ cm from the sensor. Knowing the Kinect intrinsic parameters and its rigid transformation to the screen, the displayed gaze points can be projected to the Kinect world space. The gaze vector is represented as the vector stretching the head gravity center (computed using face detection area) and the 3D gazed point. The acquisition was under SXGA and VGA resolution for RGB and depth respectively recorded at 15 fps. Fig.2b describes the acquisition process. First we grab RGB and depth information, by using the known calibration between the two sensors, a 3D textured mesh can be reconstructed. We show then, in analogy with synthetic data, some real training data used for both learning and testing.

4 Experimental results

In our experiments, we trained different forests either on real data or synthetic data. The nature of the experiment determines the training parameters.

We trained our forest $\mathcal{T}_{g_s}$ using $400k$ RGB-D synthetic gaze samples under several head pose changes. We extracted 15 patches from each sample giving 6M training data. Face depth image size is fixed to (150×150), eye RGB images to (80×70) and the size of each channel of the extracted patches for each channel is fixed to (16×16). Some training and testing parameters are fixed according to some empirical observations, $e.g.$, the maximum depth to 18 and at each node we randomly generate 400 splitting candidates with 50 thresholds giving a total number of $20k$ binary tests. At testing, we extracted a total of 30 patches from each gaze test sample with 20 regression trees. We tested our forest on 25 users from the real images database discussed previously.

4.1 Robustness to head pose and distance variations

We evaluate the gaze estimation accuracy using our trained forest $\mathcal{T}_{g_s}$ under unconstrained environment. Fig. 3a represents the global error of gaze estimation over 25 users under frontal and head changes configurations. For each user, a mean error across different gaze samples performed under two distances is computed. In frontal case, the mean error over all the users is less than $3°$ for the two directions whereas the error is less than $6.5°$ for head pose changes case. This difference in accuracy between the two configurations is directly linked to the high eye image appearances variability across head pose configuration making the trees prediction less accurate. In Fig. 3b we report the error as a function of distance from the sensor for a frontal configuration. The experiments show a mean error of $2.9°$ and $3.1°$ for θ and γ respectively at 150 cm from the sensor. At 200cm, we notified a slightly higher errors, $4.8°$ and $5.0°$ for the two directions respectively. The difference in accuracy between the two distances is related to the RGB eye images and face depth appearances which are significantly variable depending on the distance from the sensor.

4.2 Channel selection importance

To evaluate the involvement of each channel (from the two eye RGB images and face depth information) at testing time, we realized experiments using the forest $\mathcal{T}_{g_s}$ with and without depth channel and compared gaze estimation accuracy in both cases. Fig.3c illustrates the importance of depth information in our approach especially in head pose changes scenario. Gaze estimation errors are very close with and without depth information in frontal scenario whereas the error gap is approximatively $1.5°$ in head pose changes configuration proving the importance of this channel in such case. Depth information is more suitable to encode geometric similarities between data samples which represent the head pose information.

Fig.3d describes the influence of the two RGB channels (corresponding to right and left eye) on gaze estimation accuracy across different yaw angle values. These results are expected since eye appearance is very sensitive to head pose changes especially for yaw angle variation. For instance, positive values of yaw deform the left eye appearance until a complete disappearance giving high estimation errors for the two directions without the visible channel namely right eye ($i, e.$, dotted lines in Fig 3d) and reciprocally. Using our channel selection strategy introduced on the forest learning, we can quantify the involvement of each RGB channel in the final gaze estimation across head pose changes.

Fig.6 shows some clusters with low variances captured by the forest $\mathcal{T}_{g_s}$ during training step. The process is achieved with real training data.

4.3 Semi appearance-based versus appearance-based approach

Fig.4 illustrates the robustness of learning gaze in a global manifold (under head pose estimation). Instead of separating gaze and head pose as usually done, we trained two supplementary forests $\{\mathcal{T}_{g_f}, \mathcal{T}_{\mathcal{H}}\}$ on exclusively synthetic data as follows :

- $\mathcal{T}_{g_f}$: is the learned gaze estimation forest using only RGB (eye images) cues under frontal configuration exclusively. The forest is trained with the same parameters as $\mathcal{T}_{g_s}$.
- $\mathcal{T}_{\mathcal{H}}$: is the learned head pose estimation forest using RGB-D cues (face depth and face RGB images). The training parameters are fixed as done in [6] using $100k$ training data.

Fig.4a illustrates the mean error of the gaze estimation across the two direction under frontal scenario using different approaches. In red, our approach using the forest ($\mathcal{T}_{g_s}$), in green, frontal gaze estimation corrected with head pose parameters using $\{(\mathcal{T}_{g_f}, \mathcal{T}_{\mathcal{H}})\}$ and in blue, frontal gaze estimation using $\mathcal{T}_{g_f}$ corrected with an ideal head pose (driven from the OpenGl camera calibration). We noticed that errors are very close which is an expected result due to the fact that head pose parameters are not involved in the frontal scenario. Fig.4b describes the mean error in a head pose scenario. Our gaze estimation approach presents a lowest error compared to the frontal gaze estimation corrected with head pose parameters even if it is optimal. Correcting gaze estimation with head pose in a geometrical way makes the errors related to each component accumulated unlike our approach which performs a direct mapping producing an unique error for gaze estimation.

4.4 Learning with real data versus learning with synthetic data

To evaluate the realism of our rendered synthetic data and their ability to handle unconstrained gaze estimation problem, we trained a forest $\mathcal{T}_{g_r}$ on $500k$ exclusively real trai-

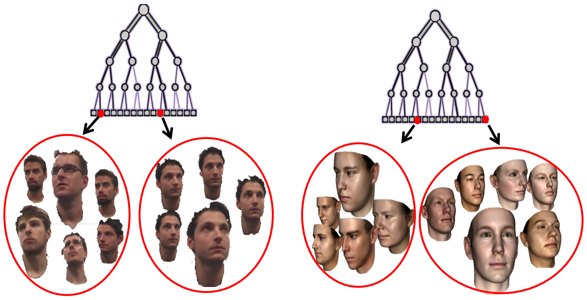
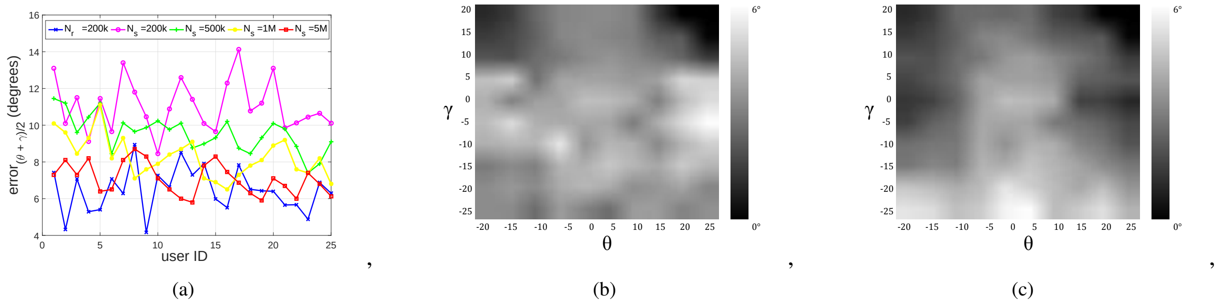


FIGURE 6 – Visualizing some clusters captured during the training step using only channel$^{(2)}$ with real and synthetic data respectively.



(a) (b) (c)

FIGURE 5 – (a) gaze estimation error (over θ and γ and over 150cm and 200cm) under learning with real and synthetic data (N_r and N_r are the number of real and synthetic training data used respectively). (b) and (c) Gaze estimation error distribution based real and synthetic learning respectively.

ning data under head pose changes extracted from the previous database (the other training parameters are kept fix as in $\mathcal{T}_{g_s}$). To achieve a comparative analysis, we trained in the same way as $\mathcal{T}_{g_s}$ different forest using different number of synthetic training data. The fact that learning forest is a very computational time task, to learn different forest in a acceptable time, we reduce the number of binary tests generated at internal nodes to $1k$ instead of $20k$ which affects considerably the estimation accuracy but yields sufficiently good results to compare different scenarios. Fig. 5a describes gaze estimation errors across different user under real and synthetic learning. A first observation can be driven by the figure, learning with the same number of training data in real and synthetic case does not perform the same accuracy which can be explained by the difference in the realism between real and synthetic data. During test, extracted patches appearance from a testing user is more closer to the gaze clusters appearances encoded using the forest $\mathcal{T}_{g_r}$. Increasing the number of synthetic training data make $\mathcal{T}_{g_s}$ increasingly close to $\mathcal{T}_{g_r}$. This can be explained by enhancing the generalization ability across unseen scenarios with more training data (We evaluated the factor between real and synthetic training producing approximatively same accuracy to $1/9$).

In Fig. 5b and Fig. 5c we illustrate gaze estimation error distribution over all 5 best testing users using $\mathcal{T}_{g_r}$ and $\mathcal{T}_{g_s}$ respectively. We can notice the importance of the error under synthetic learning in Fig. 5c for γ less than $-20°$ values resulting from the eyes closure when gazing down.

5 conclusion

In this paper, we presented a robust approach to handle gaze estimation problem in unconstrained environment using an ensemble of regression trees grouped in a single forest with high ability of generalization. To ensure the robustness, we include both RGB and depth cues as input during learning assuming a global gaze samples manifold under head pose variation. To enhance the generalization, we render an important amount of training data using a 3D morphable model with an integrated dynamic gaze model. We also, build a database with real images to evaluate the accuracy of the gaze estimation in real scenario with accuracy. Different experiments scenarios demonstrate that our approach present a great potential regarding state-of-the-art methods.

Références

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

[2] L. Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[3] R. Cappelli, A. Erol, D. Maio, and D. Maltoni. Synthetic fingerprint-image generation. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 471–474. IEEE, 2000.

[4] 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*, pages 1–4. IEEE, 2008.

[5] 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.

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

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

[8] E. D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *Biomedical Engineering, IEEE Transactions on*, 53(6) :1124–1133, 2006.

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

[10] T. Ishikawa, S. Baker, I. Matthews, and T. Kanade. Passive driver gaze tracking with active appearance models. In *Proceedings of the 11th World Congress on Intelligent Transportation Systems*, October 2004.

[11] 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*, pages 135–138. ACM, 2008.

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

[13] F. Lu, T. Okabe, Y. Sugano, and Y. Sato. A head pose-free approach for appearance-based gaze estimation. In *BMVC*, pages 1–11, 2011.

[14] 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*, pages 153–160. IEEE, 2011.

[15] 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*, pages 499–504. IEEE, 2000.

[16] 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 on*, pages 25–30. IEEE, 2012.

[17] 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.

[18] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, et al. Efficient human pose estimation from single depth images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12) :2821–2840, 2013.

[19] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike. An incremental learning method for unconstrained gaze estimation. In *Computer Vision–ECCV 2008*, pages 656–667. Springer, 2008.

[20] 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*, pages 191–195. IEEE, 2002.

[21] N. P. H. Thian, S. Marcel, and S. Bengio. Improving face authentication using virtual samples. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–233. IEEE, 2003.

[22] J.-G. Wang and E. Sung. Study on eye gaze estimation. *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, 32(3) :332–350, 2002.

[23] O. Williams, A. Blake, and R. Cipolla. Sparse and semi-supervised visual mapping with the sˆ 3gp. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 230–237. IEEE, 2006.

[24] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2015.

[25] Z. Zhu and Q. Ji. Novel eye gaze tracking techniques under natural head movement. *Biomedical Engineering, IEEE Transactions on*, 54(12) :2246–2260, 2007.

[26] 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*, volume 1, pages 1132–1135. IEEE, 2006.

[27] J. Zuo, N. A. Schmid, and X. Chen. On generation and analysis of synthetic iris images. *Information Forensics and Security, IEEE Transactions on*, 2(1) :77–90, 2007.