HAL
open science

# Unconstrained Gaze Estimation Using Random Forest Regression Voting

Amine Kacete, Renaud Séguier, Michel Collobert, Jérôme Royan

## To cite this version:

Amine Kacete, Renaud Séguier, Michel Collobert, Jérôme Royan. Unconstrained Gaze Estimation Using Random Forest Regression Voting. ACCV 13th Asian Conference on Computer Vision, Nov 2016, Taipei, Taiwan. hal-01393591

**HAL Id: hal-01393591**

**https://hal.science/hal-01393591v1**

Submitted on 7 Nov 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Unconstrained Gaze Estimation Using Random Forest Regression Voting

Amine Kacete, Renaud Séguier, Michel Collobert, and Jérôme Royan

Institute of Research and Technology B-com
Cesson-Sévigné, France

**Abstract.** In this paper we address the problem of automatic gaze estimation using a depth sensor under unconstrained head pose motion and large user-sensor distances. To achieve robustness, we formulate this problem as a regression problem. To solve the task in hand, we propose to use a regression forest according to their high ability of generalization by handling large training set. We train our trees on an important synthetic training data using a statistical model of the human face with an integrated parametric 3D eyeballs. Unlike previous works relying on learning the mapping function using only RGB cues represented by the eye image appearances, we propose to integrate the depth information around the face to build the input vector. In our experiments, we show that our approach can handle real data scenarios presenting strong head pose changes even though it is trained only on synthetic data, we illustrate also the importance of the depth information on the accuracy of the estimation especially in unconstrained scenarios.

## 1 Introduction

Automatic gaze estimation is the process of determining where the user is looking which can be represented as the point-of-regard or the visual axis. In recent years, gaze estimation has become the focus of several computer vision research according to the importance of this component in understanding the human behavior. Determining this information can be used in different areas such as Human Computer Interaction (HCI) systems, psychological and cognitive process understanding, security and monitoring systems and marketing research.

Many existing industrial solutions are commercialized and provide an acceptable accuracy in gaze estimation. These solutions often use a complex hardware such as range of infrared cameras (embedded on a head mounted or in a remote system) making them intrusive, very constrained by the user's environment and inappropriate for a large scale public use.

Current research focus on estimating gaze using low-cost devices such as a simple monocular camera relying on the analyze of the eye features and sometimes, head features extraction to infer the head pose parameters. [1] gives a very comprehensive survey about it. In this paper we present an approach based on an ensemble of trees grouped in a single forest to learn the highly non-linear

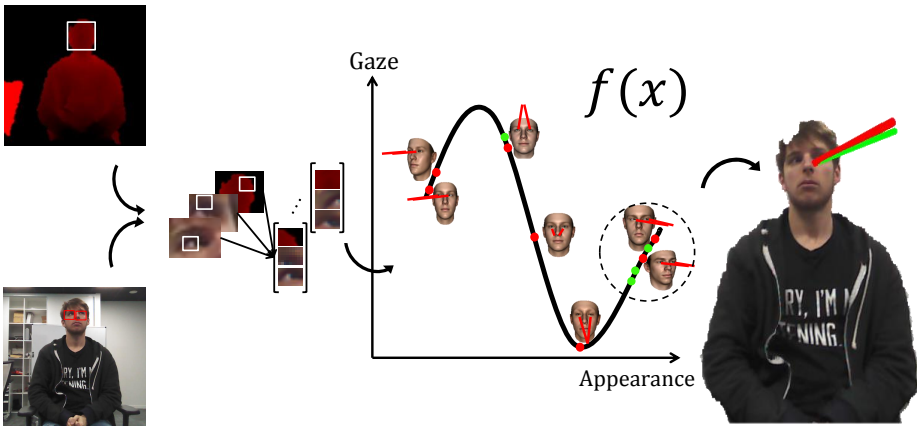


**Fig. 1.** Example of automatic gaze estimation based on our approach. We build a 3-channel global vector represented by the two RGB eye images and the face depth information using the depth sensor multimodal data, we extract a set of patches and project it through the forest represented here as the 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).

mapping function between the gaze information and the RGB eye image appearances including depth cues. To train our trees, we generate important training data with high head pose, illumination and scale variability using a statistical morphable model with an integrated parametric gaze model. At testing phase, we exploit the multimodal data of the Kinect sensor to grab the RGB and depth cues. By performing a face detection, we extract the two RGB eye images (converted to grayscale space) and the face depth information organized as a 3-channel global vector. Then we extract a set of patches so that each patch contains 3-channels extracted randomly from the global vector. We project all the extracted patches through the learned regression forest which casts votes for each patch. By clustering all the votes using a non-parametric technique, a final gaze estimate is calculated as illustrated in Fig. 1.

In our experiments we evaluate the robustness and accuracy of our approach on real data and we measure the importance of the depth cues in gaze estimation especially in highly unconstrained scenarios. The obtained results demonstrate the potential of our approach.

In the rest of the paper, we describe the related work in Sec. 2. In Sec. 3, we detail our approach and show the experimental results in Sec. 4. Sec. 5 concludes our work.

## 2 Related work

In this section, we first present the existing work related to the automatic gaze estimation then we present a brief survey of the use of synthetic data to solve computer vision problems.

### 2.1 Automatic gaze estimation

The recent gaze estimation approaches can be divided into two global categories: feature-based and appearance-based approaches.

Based on geometrical assumptions, feature-based approaches rely on extracting some discriminative and invariant facial features from the eye image such as corneal infrared reflexion, pupil center and eye corners. Using these features, a user-specific 3D eyeball model is calculated to infer the visual axis information, [2] gives details about this model. [3] used the shape of the estimated pupil through an elliptic fitting. In addition to the pupil location information, [4] used the corners locations estimated through an AAM [5] fitting and by combining the two pieces of information, calculated the center and the radius of the eyeball giving the two angles of the visual axis. To get a direct access to the 3D information of the eyeball, [6] used a stereo setup. [7] performed the same strategy with a single camera by adding a calibration step. Some work uses a depth sensor, [8] estimated the head pose parameters using a multi-template ICP, based on these parameters and a template matching approach based on elliptical fitting, the eyeball parameters can be fixed. [9] used a flexible model fitting approach to compute the head pose parameters, coupled to the pupil location information estimated using the method from [10] and a calibration step by gazing a known fixed 3D points, the visual axis can be inferred. [2] and [11] used the corneal reflexion information based on one or multiple IR light sources. These methods still require sometimes complex devices such as Infrared cameras with a very heavy constrained calibration process, and sometimes a very high resolution imaging to extract accurately the facial eye points making them difficult to use in arbitrary environment.

Our method belongs to the appearance-based approach. Unlike feature-based approach, these methods aim to learn a direct mapping function from the high dimensional eye image appearances to the low space of the gaze information. [12] trained a neuronal network using $2k$ labeled training samples. [13] collected 252 training samples to build a manifold of the local linearity of the eye appearances and estimate an unknown sample using a linear interpolation. [14] exploited the Markov model interpolation to enhance the generalization of the mapping function over unseen data such as gaze sample under head movement. [15] introduced sparse semi-supervised Gaussian process to complete the training set with unlabeled samples. [16] proposed a visual saliency maps strategy to generate training data through a video stream and used a Gaussian process regression to determine the mapping function. [17] introduced the adaptative linear regression to learn on a very sparse training set. These methods perform on frontal head pose configuration and their accuracy decrease significantly with head pose

changes. [18] proposed to separate head pose component from the global gaze estimation system by performing an initial estimation under frontal configuration assumption then compensated with the head pose parameters for the final estimation geometrically. Using the same paradigm, [19] projected the training gaze sample in frontal manifold using a frontalization step based on the head pose parameters. 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, [20] 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 still very constrained by an important computational time.

### 2.2 Synthetic data in computer vision

This last decade, machine learning techniques are considered as a very elegant way to tackle many problems in computer vision. 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. Thus, the building of high amount of labeled data is a very tedious process and synthetic data represent a promising solution as the annotation is performed automatically instead of manual labeling. [21] developed an iterative model based on Gabor-filters applied on an empty image containing some seed points to render a fingerprint training samples. [22] rendered iris image samples obtained from a 2D polar projection of a cylindrical representation of continuous fibers. [23] improved face authentication by generating multiple virtual images using simple geometric transformations. [24] 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. [25] tackled the head pose estimation problem with synthetic depth images by rendering an important amount of training data using a 3D statistical morphable model (3DMM).

In this work, we exploit the high generalization ability of the randomized regression trees by learning on a very representative rendered training data using the same 3D statistical morphable model as [25], and perform the gaze estimation.

## 3 Automatic gaze estimation with regression forest

We use randomized regression trees to estimate the two angles $(\theta, \gamma)$ of the gaze vector $\overrightarrow{g}$ from the RGB and depth cues combined on 3-channel patches. In Sec. 3.1, we provide some background on regression trees, then we detail the training and testing step in Sec. 3.2 and Sec.3.3 respectively. In Sec. 3.4 we illustrate how we generated data for trees learning.

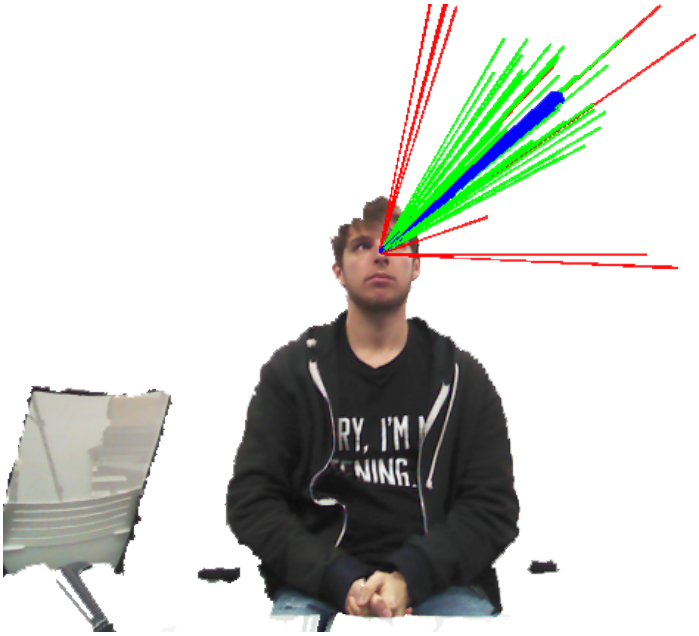

**Fig. 2.** test instance example: the selected votes by mean-shift filtering are represented in green, the non-informative leaves responses in red and the final estimation in blue.

### 3.1 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 space. Introduced by [26], randomized trees deal with different tasks such as classification [27–29], regression [24, 30, 31] and density estimation [32, 33].

Regression forest is an ensemble of trees 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. To achieve high generalization, the trees are trained in a decorrelated way by introducing randomness in both the training data provided for each tree and the set of the binary tests.

### 3.2 Training

We trained each tree T in the forest $\mathcal{T} = \{T_k\}_{k=1:N_T}$ in a supervised way using a set of annotated patches $\{\mathcal{P}_i = (\mathcal{I}_i^c, g_i)\}_{i=1:N_\mathcal{P}}$ 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 eyes images, and the depth values extracted from the face.
- $g_i$ represents the output gaze vector represented with two component $(\theta, \gamma)$.

Starting from the root, at each non-leaf node, we define a simple binary test t:

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

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 $t^*$ thats maximizes the purity of the data clustering. Maximizing the clustering purity 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:

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

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

$$E = \log|\Sigma(\mathcal{P})| - (w_L \log|\Sigma(\mathcal{P}_\mathcal{L})| + w_R \log|\Sigma(\mathcal{P}_\mathcal{R})|) \tag{2}$$

where $|\Sigma(\mathcal{P})|$ represents the determinant of the covariance matrix $\Sigma$ of the random variable $g$.

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.

### 3.3 Testing

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

$$p(g|\mathcal{P}_i) = \frac{1}{N_T} \sum_t p(g|l_t(\mathcal{P}_i)) \tag{3}$$

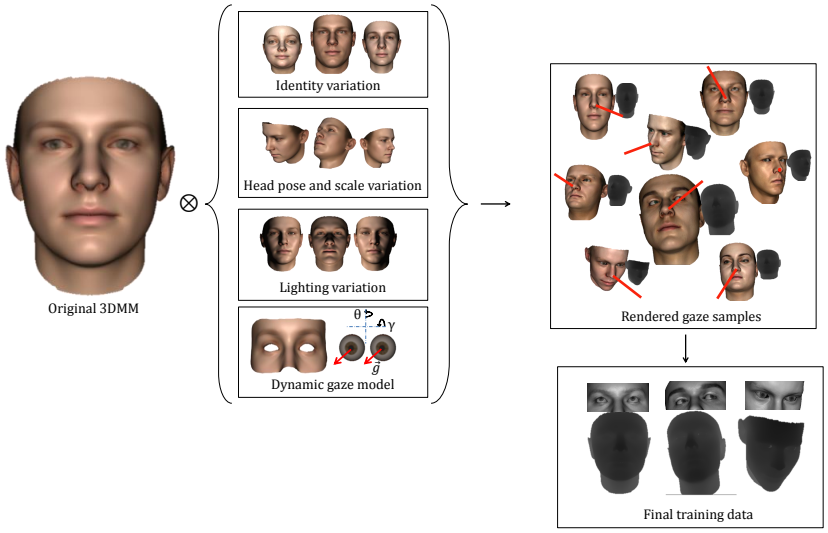


**Fig. 3.** Training data generation. We based our data generation on the 3D morphable model from [34], by introducing some variabilities such as identity (using shape and texture principal components respectively), head pose changes ( using OpenGL camera with different rigid transformations), illumination (using different light intensities and directions) and an integrated parametric gaze model (represented by two global textured spheres). We obtain the final training data with the correspondent RGB-D images and gaze sample as annotation illustrated in red line.

All the estimations corresponding to the extracted patches are regrouped in 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. Fig. 2 shows an example of the final estimation, the green ones represent the votes casted by the forest which are selected by the mean-shift. The red lines corresponds some casted votes with a high variance discarded by the mean-shift. The final estimate is given by the blue line corresponding to the centroid of the selected votes.

### 3.4 Training data generation

To provide a very representative training dataset, we use the 3DMM from [34] to render the samples. This model is built from around 200 scans of human faces, it contains a very high mesh density including the face, frontal neck and ears. The shape and texture of the model are represented as a linear combinations of 199 components. They can be deformed according to the following equations:

$$\mathcal{A} = \mathcal{A}_0 + \mathcal{M}_{\mathcal{A}}\alpha \tag{4}$$

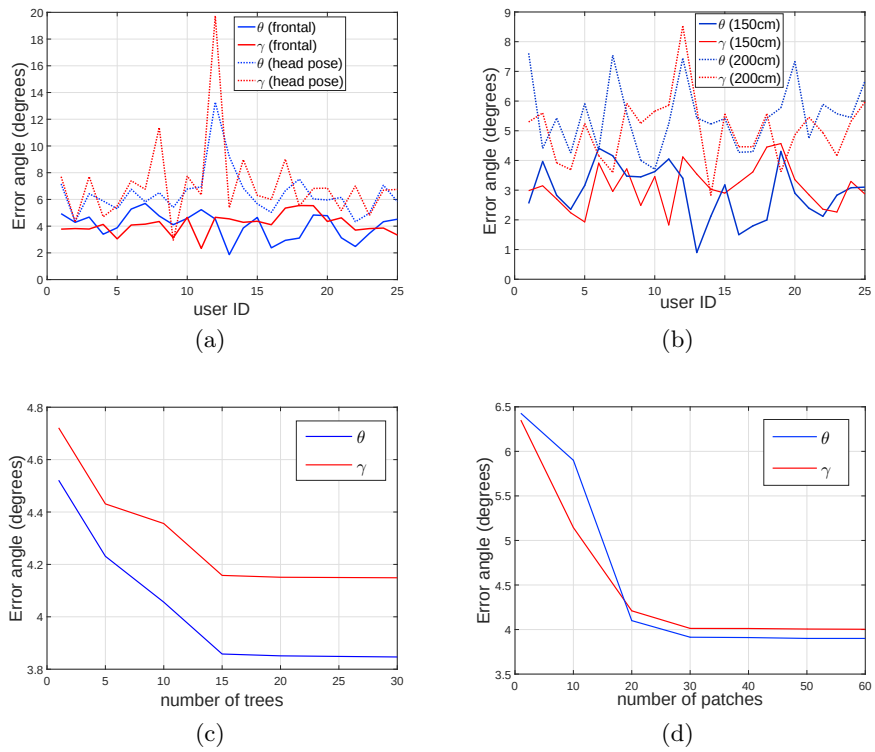


**Fig. 4.** Testing our forest accuracy learned with synthetic data on real annotated images. a) The mean error for the two gaze directions under frontal and head pose changes. b) The mean error of the two gaze directions under two distances from the sensor. c) The global mean error (over 25 participants as a function of the number trees when the number of patches is fixed to 30). d) The global mean error as a function of the number of patches extracted when the number of trees is fixed to 15.

where $\mathcal{A}$ can denote the generated texture or shape respectively. $\mathcal{A}_0$ denotes the mean, $\mathcal{M}$ represents the basis components perturbed with parameters $\alpha$.

To generate variability in the face identity we perturb the first 50 basis components of the shape and texture by $\pm 1.5$ of the standard deviation of each mode. To render images in different head pose configurations, we apply random rigid transformation on the model, the rotations spans $\pm 60°$ for yaw and $\pm 40°$ for pitch. For the scale, we translate the model along the $z$ axis within 200 cm range.

Unfortunately, the basis components related to the shape and the texture of this model do not monitor the gaze direction. To integrate a parametric gaze model to the 3DMM able to generate different gaze direction instances, we decided to remove all the vertices related to the eye regions, and we place two spheres as eyeballs instead. We fix the diameters to the human average eyeball

namely 25 mm. We use different textures for the eyeballs to handle the iris appearance variability. Moreover, to control the eyelids movements resulting from the gazing up and down, we introduce a linear translation for each vertex surrounding the eye regions. 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 sample, we generate a virtual 3D point on which the two eyeballs turn toward, the gaze information angles can be easily calculated knowing the location of the eyeballs centers. Fig. 3 shows the different steps applied to generate gaze samples.

## 4 Experimental results

**Training dataset** To train the regression trees, we used $200k$ synthetic RGB-D samples. We extracted 30 patches from each sample giving $6M$ training data. After scaling the face depth image to $(150 \times 150)$ and the eyes rgb images to $(80 \times 70)$, the size of each channel of the extracted patches is fixed to $(16 \times 16)$. The trees parameters are fixed according to some empirical observation, *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.

**Testing dataset** To evaluate the performance of our algorithm on realistic data, we built our own gaze database using 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 the screen. The subject performed 4 scenarios,gazing with a fixed head about $d_0 = 150$ cm from the sensor, gazing with same distance $d_0$ under head pose changes and the two others scenarios are performed 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 vector stretching the head gravity point (computed using face detection area) and the 3D gazed point. The RGB-D images have a resolution of $(1280 \times 960)$ and $(320 \times 240)$ pixels respectively recorded at 15 fps.

**Testing results** Some parameters control the performance of our method at the test time. Fig. 4a represents the global error of the estimation (for both horizontal $\theta$ and vertical $\gamma$ gaze angles) over 25 users from the database discussed previously under frontal and head changes configurations. For each user, a mean error across different gaze samples performed under two distances is calculated. In frontal case, the mean error over all the users is less than $3°$ for the two directions respectively 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. 4b 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 $\theta$ and $\gamma$ respectively at 150 cm from the sensor. At 200 cm, we notified a slightly higher errors, $4.8°$ and $5.0°$ for the two directions

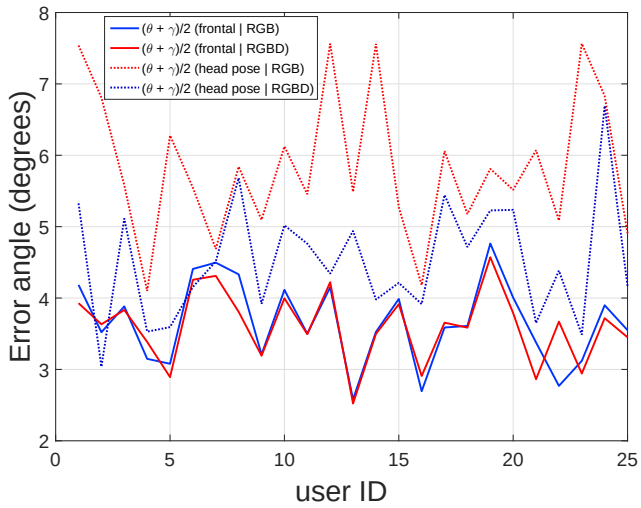


**Fig. 5.** The importance of depth channel in gaze estimation accuracy with our approach.

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. Fig. 4c and 4d illustrate the variation of the mean errors over all the testing user under forest size and extracted patches number variation. In Fig.4c the errors decrease by increasing the number of trees, they are reduced by approximately 15% compared to the initial value (from 4.5° to 3.8° and 4.7° to 4.1° for the two directions receptively) which is the result of output smoothing by different trees. We noticed that, using more than 15 trees does not perform more precision, so we fix the optimal forest size to 15. The number of patches extracted in the testing step is fixed to 30 according to Fig. 4d showing that the errors decrease approximatively by 40% (from 6.4 to 3.8 and 6.4 to 3.9 for the two directions). This behavior can be explained by the fact that trees get more information about the input which consequently gives more accurate estimations. To evaluate the importance of depth cues in our gaze estimation system, we performed our estimation with and without this information during the test under frontal and head pose changes, Fig. 5 shows the result. For the frontal configuration, we noticed that the depth doesn't enhance the estimation accuracy while difference in errors reach approximatively 2° under head pose changes. This result is expected since the depth cues, intrinsically, encodes more information related to the head pose variations than RGB cues giving better results.

Fig.6 illustrates the distribution of the mean square error of two gaze directions across $\theta$ and $\gamma$ variations. We can distinguish 3 regions as follows:

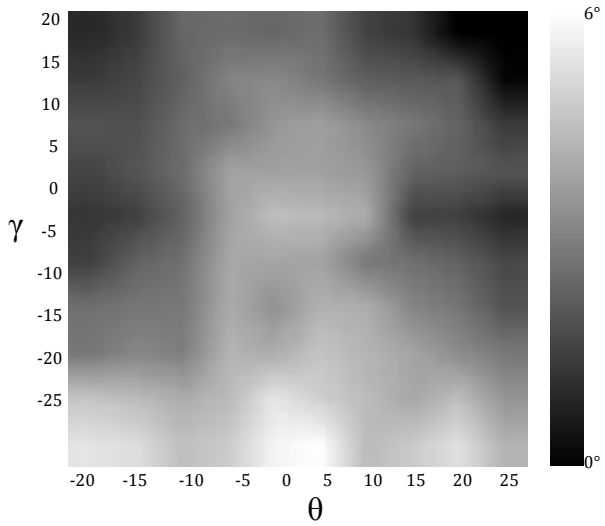


**Fig. 6.** Mean gaze error distribution over 25 participants across different gaze directions.

| Method | $\theta$ error | $\gamma$ error |
|---|---|---|
| Jianfeng et al., | 5.53° | 4.51° |
| Our method (worst participant result) | 5.78° | 5.81° |
| Our method (best participant result) | 2.76° | 2.93° |
| **Our method (mean over 25 participants)** | 3.65○ | 3.88° |

**Table 1.** Comparison of our approach to the method in [9].

- $\gamma < -20°$ represents the highest error range ). These $\gamma$ values correspond to the eyes closure making the eye image appearances very similar even if $\theta$ is varying which produces bad gaze estimations. Furthermore, our parametric gaze model performs a linear shifting on the eyelid vertices to cover the new eye shape and stretches the original eyelid texture to cover the new texture giving a rough approximation of the real eye appearance. Our choice of such gaze model is strongly constrained by the 3DMM topology.
- $|\theta| < -7°$ describes a region with a relatively important error. Our forest is weakly discriminative with straight gazing samples under large distances. In addition, we noticed, for some users, an important error for upward gazing configuration($\gamma > 10°$ and $\theta < 5°$) which can be explained by an elliptical deformation of the high part of the eyes. The fact that, this deformation is very person-specific and our parametric model performs the same deformation over the different face shapes generated by the 3DMM, the forest gives less accurate results.
- $\gamma > -20°$ and $|\theta| > 5°$ covers the range of good gaze estimation (error less than 4°) which represents more than 50% of the total area. The appearance of the patches extracted from these gaze samples are very discriminative, in

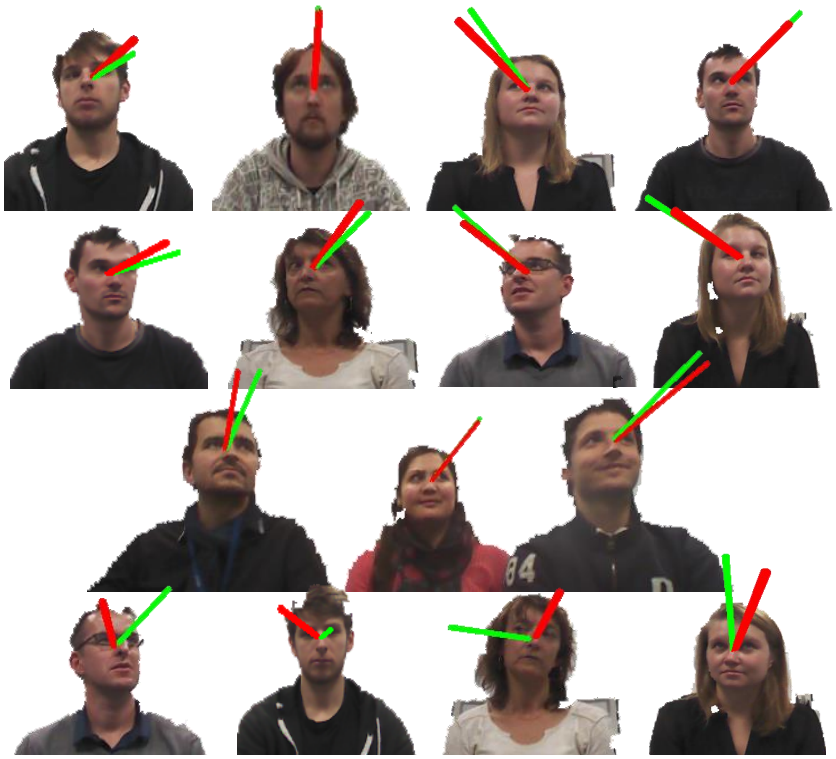

**Fig. 7.** Some example of automatic gaze estimation based on our approach. First and second row illustrate some successful estimation under frontal and head changes configurations respectively. Third row shows good estimation in a multi-users scenario. Last row describes some gaze estimations failure using our approach.

addition, for these configurations, our synthetic training data present a very high realism.

Table. 1 compares our approach to the method from [9]. In the experiments conducted in [9] the gaze errors are computed for each eye, to get a direct comparison, we reported the mean of these errors over the two eyes. Note the improvement of our approach in accuracy overs 25 participants. Fig. 7 shows some qualitative examples for successful and failure estimations.

## 5 Conclusion

In this paper, we have proposed an approach based on regression forest trained on important amount of synthetic data to handle unconstrained gaze estimation (under head pose changes, illumination variation and large user-sensor distances). To generate the training data, we used a 3D morphable model with an integrated parametric gaze model allowing us to generate different gaze sample

under identity, head pose, scale and illumination variations. We demonstrated that adding depth information performs better results for gaze estimation under high head pose changes and large user-sensor distance configurations. By establishing the gaze errors distribution we validate our integrated gaze model used for training data generation despite its linear aspect achieving state-of-the-art performance.

## References


1. Hansen, D.W., Ji, Q.: In the eye of the beholder: A survey of models for eyes and gaze. TPAMI (2010)
2. Guestrin, E.D., Eizenman, M.: General theory of remote gaze estimation using the pupil center and corneal reflections. Biomedical Engineering, IEEE Transactions on **53** (2006) 1124–1133
3. Wang, J.G., Sung, E.: Study on eye gaze estimation. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **32** (2002) 332–350
4. Ishikawa, T.: Passive driver gaze tracking with active appearance models. (2004)
5. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. TPAMI (2001)
6. Matsumoto, Y., Zelinsky, A.: 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, IEEE (2000) 499–504
7. Chen, J., Ji, Q.: 3d gaze estimation with a single camera without ir illumination. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE (2008) 1–4
8. Bär, T., Reuter, J.F., Zöllner, J.M.: Driver head pose and gaze estimation based on multi-template icp 3-d point cloud alignment. In: Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on, IEEE (2012) 1797–1802
9. Jianfeng, L., Shigang, L.: Eye-model-based gaze estimation by rgb-d camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2014) 592–596
10. Timm, F., Barth, E.: Accurate eye centre localisation by means of gradients. In: VISAPP. (2011)
11. Zhu, Z., Ji, Q.: Novel eye gaze tracking techniques under natural head movement. Biomedical Engineering, IEEE Transactions on **54** (2007) 2246–2260
12. Baluja, S., Pomerleau, D.: Non-intrusive gaze tracking using artificial neural networks. Technical report, DTIC Document (1994)
13. Tan, K.H., Kriegman, D.J., Ahuja, N.: Appearance-based eye gaze estimation. In: Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on, IEEE (2002) 191–195
14. Hansen, D.W., Hansen, J.P., Nielsen, M., Johansen, A.S., Stegmann, M.B.: Eye typing using markov and active appearance models. In: Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on, IEEE (2002) 132–136
15. Williams, O., Blake, A., Cipolla, R.: Sparse and semi-supervised visual mapping with the s^ 3gp. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 1., IEEE (2006) 230–237

16. Sugano, Y., Matsushita, Y., Sato, Y.: Calibration-free gaze sensing using saliency maps. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 2667–2674
17. Lu, F., Sugano, Y., Okabe, T., Sato, Y.: Inferring human gaze from appearance via adaptive linear regression. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 153–160
18. Lu, F., Okabe, T., Sugano, Y., Sato, Y.: A head pose-free approach for appearance-based gaze estimation. In: BMVC. (2011) 1–11
19. Mora, K.A.F., Odobez, J.M.: Gaze estimation from multimodal kinect data. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, IEEE (2012) 25–30
20. Zhang, X., Sugano, Y., Fritz, M., Bulling, A.: Appearance-based gaze estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4511–4520
21. Cappelli, R., Erol, A., Maio, D., Maltoni, D.: Synthetic fingerprint-image generation. In: Pattern Recognition, 2000. Proceedings. 15th International Conference on. Volume 3., IEEE (2000) 471–474
22. Zuo, J., Schmid, N.A., Chen, X.: On generation and analysis of synthetic iris images. Information Forensics and Security, IEEE Transactions on **2** (2007) 77–90
23. Thian, N.P.H., Marcel, S., Bengio, S.: Improving face authentication using virtual samples. In: Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. Volume 3., IEEE (2003) III–233
24. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56** (2013) 116–124
25. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: CVPR. (2011)
26. Breiman, L.: Random forests. Machine learning **45** (2001) 5–32
27. Marée, R., Wehenkel, L., Geurts, P.: Extremely randomized trees and random subwindows for image classification, annotation, and retrieval. In: Decision Forests for Computer Vision and Medical Image Analysis. Springer (2013) 125–141
28. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. TPAMI (2011)
29. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: CVPR. (2005)
30. Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression forests for efficient anatomy detection and localization in ct studies. In: Medical Computer Vision Workshop. (2010)
31. Kacete, A., Seguier, R., Royan, J., Collobert, M., Soladie, C.: Real-time eye pupil localization using hough regression forest. In: Applications of Computer Vision, 2016.(WACV 2016). Proceedings. Sixth IEEE Workshop on, IEEE (2016)
32. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Twentieth Annual Conference on Neural Information Processing Systems (NIPS'06), MIT Press (2007) 985–992
33. Ram, P., Gray, A.G.: Density estimation trees. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2011) 627–635
34. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3d face model for pose and illumination invariant face recognition. In: Advanced Video and Signal Based Surveillance. (2009)