Appearance-Based Gaze Tracking: A Brief Review

Jiaqi Jiang[1], Xiaolong Zhou[1,2(✉)], Sixian Chan[1], and Shengyong Chen[1,3]

[1] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
zxl@zjut.edu.cn
[2] College of Electrical and Information Engineering, Quzhou University, Quzhou, China
[3] School of Computer Communication and Engineering, Tianjin University of Technology, Tianjin, China

Abstract. Human gaze tracking plays an important role in the field of Human-Computer Interaction. This paper presents a brief review on appearance-based gaze tracking. Based on the appearance of human eyes, input features can be classified into three categories according to the different ways of extracting human eyes features, namely, complete human eye image, pixel-based feature and 3D reconstruction image. The estimation process from human eye feature to fixation point mainly uses different mapping functions. In this paper, common mapping functions and related algorithms are described in detail: *k*-nearest neighbor (KNN), random forest (RF) regression, gaussian process (GP) regression, support vector machines (SVM) and artificial neural networks (ANN). This paper evaluates the performance of these gaze tracking algorithms using different mapping functions. Based on the results of the evaluation, potential challenges are summarized and the future directions of gaze estimation are prospected.

Keywords: Gaze tracking · HCI · Appearance-based · Mapping

1 Introduction

The eyes are one of the most important sensory organs in the human body, which is the inevitable result of the long evolution of life to the advanced form. More than 90% of external information is obtained through the eyes. Eye gazing plays an important role in nonverbal communication as well as Human-Computer Interaction (HCI) [1–3]. Gaze tracking can be used as an analytical tool in HCI to make the interaction between human and computer more simple, natural and efficient. Gaze tracking plays an important role in many applications, including marketing and consumer research [4], immersive VR research [5], education research [6], et al.

There have been an increasing number of recent methods proposed for gaze tracking, which can be roughly classified into two major categories: model-based and appearance-based methods. The former calculates the specific geometric eye model to

H. Yu et al. (Eds.): ICIRA 2019, LNAI 11745, pp. 629–640, 2019.
https://doi.org/10.1007/978-3-030-27529-7_53

estimate gaze direction relying on invariant facial features such as pupil center [7], eye corners [8] and corneal infrared reflection [9]. The latter extracts input features from the human eye appearance images and establishes a mapping relation to realize gaze estimation. Common input eye features can be roughly divided into three categories: complete human eye images [10, 11], pixel-based features [12–14], and 3D reconstructed images [15]. The model-based gaze tracking methods require sophisticated hardware which may be composed by infrared light and high-definition cameras. Such methods are more suitable for controlled environments, e.g., in the laboratory, rather than in daily entertainment scenes. In contrast, the appearance-based methods usually only need a single camera to capture the user eye images. Certain eye features are generated from the complete eye images, and then a gaze mapping function is learned that maps the eye image to the gaze direction. Common eye features include a complete human eye image and the pixel-related information extracted from it, including color, gradient, light histogram, etc. Such a mapping function can be learned using various regression techniques, including *k*-Nearest Neighbor (KNN) [15–17], Random Forest (RF) regression [18–21], Gaussian Process (GP) regression [23–26], Support Vector Machines (SVM) [15, 27–29] and Artificial Neural Networks (ANN) [31–38].

The rest of this paper is organized as follows. Section 2 presents the common mapping functions and their related gaze tracking methods in detail: KNN, RF, GP, SVM and ANN. The advantages and disadvantages of these functions are briefly described. In Sect. 3, the performance of different mapping functions on appearance-based gaze tracking are evaluated. The potential challenges are summarized and the research directions of gaze estimation are prospected.

2 Appearance-Based Gaze Tracking

The estimation process from human eye feature to fixation point (gaze direction) mainly adopts different mapping functions, including establishing regression function and different clustering methods. The commonly used mapping functions mainly includes KNN, RF regression, GP regression, SVM, ANN and the convolutional neural network expanded on it.

2.1 *K*-Nearest-Neighbor

KNN is to calculate the distance between different eigenvalues to realize classification. Given the training set and the label, KNN first compares the characteristics of the input test data with the corresponding characteristics in the training set, and then calculates the distance between the training data and the test data, and finally gets the k points with the minimum distance (that is, the most similar) according to the distance ordering. Most of the categories of k points are the categories of this point.

The KNN algorithm needs to calculate the Euclidean distance between training data, so it will lead to excessive computational complexity in the case of large sample set calculation, and this algorithm is suitable for the case of small sample.

Zhang et al. [16] first extracted three low-dimensional features, including the color opponency, gray scale intensities and direction information. Secondly, the feature

vector was gotten by normalized averaging the three features information, and minimum redundancy maximum relevance (MRMR) feature selection was used to reduce high dimensional image data to low dimensional feature vector. Finally, a KNN classifier with $k = 13$ was used to learn the mapping from image features to gaze direction. The reason why this method could use KNN algorithm was that it directly divided the gaze direction into 13 categories, and the value of k was not large, so the regression problem was changed to the classification problem.

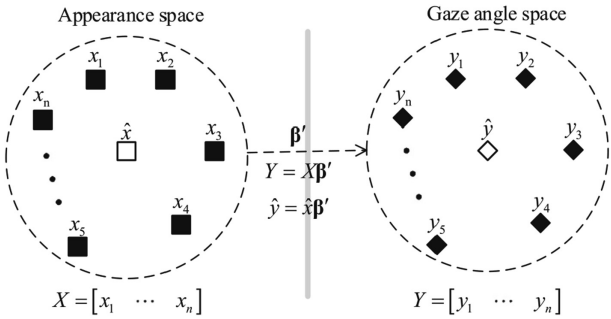

Fig. 1. The overall framework of neighbor regression [17]

As shown in Fig. 1, Wang et al. [17] proposed a gaze estimation method by combining with neighbor selection and neighbor regression. To find the closest set of samples, neighbor selection used features to achieve this goal, including head pose information, pupil center feature and eye appearance feature. This operation took up a large part of the total execution time, however, k-neural network adopted k-d tree structure in each feature space, it could achieve faster query speed. The neighbor regression improved the previous gaze estimation based on k-nearest neighbors that considered the correlation between samples and gaze angles to build models between the appearance space and gaze angle space. To save time, Weighted Least Squares Regression (WLSR) is selected as the regression method. Wood et al. [15] reconstructed the human eye image by scanning the 3D face with high resolution, used the KNN algorithm to obtain the gaze vector by matching the rendered human eye images to the images from MPIIGaze dataset.

From the existing KNN-based gaze tracking methods, it is worth considering how to balance the value of k and the time consumption. In contrast, KNN algorithm is more suitable for the case of fewer samples.

2.2 Random Forest Regression

RF is a classifier that contains multiple decision trees. It uses random samples to generate decision trees, called random decision tree. RF extracts some samples from the original data that need to be put back to generate the sample set and repeats the above steps to generate multiple sets. A decision tree is generated for each sample set, however, there is no correlation between decision trees. After obtaining the forest, a

new test data is input into each decision tree, and the corresponding output value is generated. The output of the test data is the average output of each tree in the forest. RF is widely used in head pose estimation, feature selection and recommendation system. In recent years, RF has been employed in gaze tracking.

Obviously, RF can process high-dimensional data without feature selection, and the training speed is fast because it can operate in parallel. In order to solve the problem of head posture under deep illumination, Wang et al. [18] added the depth feature to the traditional line-of-sight estimation based on appearance, and applied the RF regression with cluster-to-classify node splitting rules. The concrete operation steps are divided into two steps: generating RF and predicting RF.

Generating RF

Assuming that x and y were the i-th depth feature vector and two-dimensional line of sight vector respectively, namely input variable and output variable, N represented the number of calibration point, then the entire training data set was represented by $\{x_i, y_i\}_{i=1}^{N}$. The forest contained M tree $T = \{T_1, T_2, \ldots, T_M\}$. Before each node splitting, computing feature local density ρ_m and distance measure δ_m, and recognizing cluster centers sorted by $\rho_m > \min(\rho)$ and $\delta_m > \min(\delta)$. Then, cluster $\{C_1, C_2, \ldots, C_k\}$ was completed by assigning the remaining feature points to the same cluster as its nearest high-density neighbor. Finally, the node was segmented by computing

$$f = \min_{w_k} \|w_k\|_2 + Z \sum_{i=1}^{m} \left(\max(0, 1 - d_i^k w_k^T x_i)\right)^2 \tag{1}$$

where w_k was the k-th cluster weight vector, Z was the penalty parameter. If $x_i \in C_k$, $d_i^k = 1$, otherwise, $d_i^k = -1$. Repeating the above steps until all nodes were partitioned to produce each random tree, then getting the regression value y_{pre}^i of current tree T_i.

Predicting RF

Entering the test sample x_{test} into each random tree, selecting the optimal splitting variables w_k and splitting points K, the gaze vector of the test set was the average of the regression values for each tree.

RF processed high-dimensional data without feature selection, and there was a large number of missing data, it had a strong anti-interference ability. The experimental results were the best when the number of features equaled to 160. Therefore, in the feature extraction stage, functions should be extracted as many as possible, however, with the increase of the number of features, the experimental results would also become worse. Excessive features may lead to redundancy (feature correlation is too high, part of consumption performance), noise (part of features have a negative impact on the predicted results), overfitting and other problems.

Sugano et al. [19] proposed a gaze estimation method using RF regression. Because this method collected the largest and fully calibrated multi-view gaze dataset and performed a 3D reconstruction eye images to build the input vector, RF regression could handle large-scale regression problems at a lower computational cost. Kacete et al. [20] used RF regression to estimate the gaze vector from the depth information with the face information. This method could handle real data scenarios presenting

strong head pose changes. In general, the RF method was suitable for processing high-dimensional data. It could do parallel processing as well as the training speed was relatively fast. However, there would be overfitting in some regression problems with high noise. Huang et al. [21] studied gaze tracking on tablets. The various practical factors were performed extensive evaluation by the baseline algorithm which was based on multi-level HOG feature and RF regressor.

2.3 Gaussian Processers

GP [22] is a collection of random variables indexed by time or space, and the distributions of various derived quantities can be obtained explicitly. GP is different from the general regression algorithm in that the general regression algorithm is given the input x to get the corresponding output y, while the GP is to get the distribution of the function $f(x)$. The advantage of the GP is that it can get not only the estimate of the output, but also the confidence interval of the estimate.

For the training set D: (X, Y), let $f(X) = Y$, then the vector $f = [f(x_1), f(x_2), \ldots, f(x_n)]$ can be obtained. The test set of prediction x_i is defined as X^*, and the corresponding predicted value is f^*. According to the Bayes formula:

$$p(f^*|f) = \frac{p(f|f^*)p(f^*)}{p(f)} = \frac{p(f, f^*)}{p(f)} \tag{2}$$

the joint probability distribution between samples in the training set f is calculated first, and then the posterior probability distribution of f^* is calculated according to the prior probability distribution of the prediction set f^*. However, GP is not suitable for large data sets. For data sets with sample size N, the complexity of the traditional GP regression can reach $O(N^3)$.

Wojke et al. [23] generated a lower dimensional gaze manifold using GP latent variable model with the eye patches and the corresponding gaze points. In this particular application, only two potential dimensions are used to capture relevant information. Standard GP regression was used to establish the mapping relationship between the screen coordinates and the two-dimensional feature space, and the mapping relationship was used to generate eye-patches for each gaze point in the screen coordinates. Finally, the gaze point was estimated by non-linear optimization. Williams et al. [24] introduced a semi-supervised GP regression model to learn a mapping with only partially labelled training data. This model combined probability filters and the ability to learn from semi-supervised data simplified the process of collecting training data. In order to reduce the data set, Ferhat et al. [25] used an average eye image of the subjects for each calibration target as input to train GP estimator. Sugano et al. [26] also used the average human eye image as input. The gaze probability maps were formed by clustering with the saliency maps instead of gaze points, and the mapping from the average human eye image to the gaze probability graph was constructed by the GP.

In summary, the predicted value is probability of GP. The confidence interval is calculated and then a prediction of a specific field of interest is obtained based on the relevant information. However, this method does not support large data sets, also is not sparse, so before using such methods often have to deal with these two problems.

2.4 Support Vector Machines

SVM solves linear separable problems first, however, in some cases it cannot find linearly separable partition planes. Therefore, SVM needs to use the well-known "nuclear mechanism" to map these data into high-dimensional space, and transforms the original low-dimensional nonlinear regression problem into a high-dimensional linear regression problem. The performance of SVM depends on the construction of kernel function and the selection of corresponding parameters. However, for data sets with sample size N, the time complexity of SVM regression is $O(N^3)$, which greatly limits the scalability of large data sets.

For the general regression problem, $f(x)$ is learned from the training sample to make it as close as possible to y and the loss is zero only when $f(x)$ is completely the same as the real value. However, SVM regression can tolerate the deviation of $f(x)$ and y with the maximum of ξ, and only when the deviation value is greater than e, the loss will be calculated. If the training sample is between the interval bands with a width of 2ξ, the prediction could be considered correct.

Huang et al. [27] used iris center as input features of the SVM regression to build up mapping function. Unlike the former, Zhu et al. [28] built up an approximate generalized gaze mapping function from the pupil-glint vector and 3D eye position to the screen coordinate. Wu et al. [29] located the eye region by modifying the characteristics of active appearance model, and used SVM to classify the five gaze directions. Chuang et al. [30] defined a feature descriptor by the locations and scales of face parts. Then the feature descriptor was supplied to an SVM gaze classifier to get the gaze direction. In general, SVM is robust, and a small number of support vectors can determine the final result, such as the iris center as input. In the actual gaze tracking, it is obviously not affected by different users. However, with the increase of training samples, this method cannot be implemented and it is difficult to solve the problem of multi-classification.

2.5 Artificial Neural Network

ANN is an operational model, which abstracts the neural network of human brain from the perspective of information processing and forms different networks by connecting a large number of neurons in different ways. Each neuron represents a specific activation function, and the connection between two neurons has a weight of w_i. The output of the network varies according to the connection mode, weight value and excitation function of the network. The network itself can form an approximation to an algorithm or function.

Baluja and Pomerleau [31] developed an ANN-based gaze tracking system that was tested extensively on three architectures epochs. This system could achieve an accuracy of 1.5° that only used images of the eyeball and cornea as input. However, in order to get more information about the head pose, it took three minutes to extract the high pixel images. Yu et al. [32] proposed a method of gaze tracking based on BP neural network, and used particle swarm optimization algorithm to optimize the regression model of connection weight and threshold values. This method could accurately extract the

eye-gaze features when the image pickup requirements were low. However, this method still could not achieve the full range of gaze tracking.

Convolutional Neural Network (CNN) is a special deep artificial neural network. On the one hand, the connections between neurons are local connections. For example, each layer node of BP network is a one-dimensional ordering state, and the network nodes between layers are completely connected. It's a simple one-dimensional convolutional network from a fully connected to a locally connected, and if it extends to two dimensions, it's a CNN. On the other hand, the weights between neurons on the same feature graph are the same. At present, with the development of neural network, the gaze-tracking method based on convolution network is more popular. Sewell and Komogortsev [33] used a multimodal CNN model to learn the mapping from the input features to gaze angles in the normalized space, the input features including both eye image and head pose information. As shown in the Fig. 2, which was the architecture of the proposed multimodal CNN. This model used LeNet network architecture, including two convolution layers, two max pooling layers and a fully connected layer, in which the head pose vectors were added to the output of the fully connected layer.

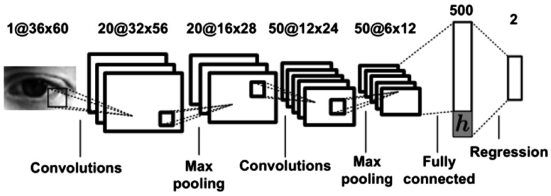

Fig. 2. Architecture of multimodal CNN [33]

Zhang et al. [34] used CNN to encode face images and imposed spatial weights on feature maps to flexibly suppress or enhance the information of different face regions. Cheng et al. [35] used the eye images and head pose information as input to complete judgement of the notion of two eye asymmetry. Zhang et al. [36] focused on the effect of variable head pose and proposed a novel branched CNN architecture that improved the robustness of gaze classifiers without increasing the computational cost. Palmero et al. [37] used face, eye region and face landmarks as separate information flows in CNN to estimate gaze in static images. It was the first time that the gaze dynamic characteristics were considered in the method. The learning features of all frames were input into a many-to-one recurrent module sequentially, and the 3D gaze vector of the last frame were predicted. Fischer et al. [38] recorded a new dataset of different head postures in order to improve the robustness of gaze estimation, and applied semantic image inpainting to the area covered by glasses to eliminate the obtrusiveness of the glasses and built a birdge between training and test images. Yu et al. [39] introduced a constrained landmark-Gaze model to get the relation of eye landmark locations and gaze directions.

In general, the training process of CNN model is long, but its accuracy and robustness are better than most standard machine learning algorithms.

Table 1. Summary of the appearance-based methods which are simply classified by means of mapping for facilitating user's access. Head pose shows whether the method has head-free movement or not, symbol ≈ means that the method allows a range of head movement, symbol ✓ means the method with fixed head pose, symbol—xmeans the method with free head movement.

Mapping	Reference	Year	Accuracy	Head pose	Calibration	Training dataset
KNN	Zhang et al. [16]	2011	/	✓	No	17 subjects
	Wang et al. [17]	2017	7.5° on SynthesEyes, 4.8° under UnityEyes	–	No	SynthesEyes, UnityEyes
	Wood et al. [15]	2016	9.95°	–	No	UnityEyes
RF regression	Wang et al. [18]	2016	1.53°	–	Yes	6 subjects with 25 training points
	Sugnao et al. [19]	2014	Average 6.5°	–	No	50 subjects with 10 grids
	Kacete et al. [20]	2016	Average 3.8°	–	No	200 k synthetic RGB-D samples
	Huang et al. [21]	2017	/	–	No	51 subjects with 35 training points
GP	Wojke et al. [23]	2016	/	≈	No	400 samples
	Blake et al. [24]	2006	0.83°	–	Yes	videos
	Ferhat et al. [25]	2014	<1.5°	✓	Yes	12 subjects
	Sugano et al. [26]	2013	3.5°	✓	No	7 subjects with 80 short clips
SVM	Huang et al. [27]	2011	0.4°	✓	Yes	/
	Zhu et al. [28]	2006	1.5°	≈	No	2757 samples
	Wu et al. [29]	2014	/	✓	No	15 subjects with 800 images
	Chuang et al. [30]	2014	/	✓	No	videos
ANN	Baluja et al. [31]	1993	Average 1.7°	≈	Yes	2000 images
	Yu et al. [32]	2016	/	✓	Yes	50 subjects with the 15 training points
	Sewell et al. [33]	2010	<3.68°	≈	Yes	5 subjects with the 50 training points
CNN (special ANN)	Zhang et al. [34]	2016	4.8° on MPIIGaze, 6.0° on EyeDiap	–	No	MPIIGaze, EyeDiap
	Cheng et al. [35]	2018	Average 5.0°	–	No	Modified MPIIGaze, UT Multiview, EyeDiap

(*continued*)

Table 1. (*continued*)

Mapping	Reference	Year	Accuracy	Head pose	Calibration	Training dataset
	Zhang et al. [36]		7.74°	–	No	MPIIGaze
	Palmero et al. [37]	2018	Average 6.2°	–	No	EyeDaip
	Fischer et al. [38]	2018	4.3° on MPIIGaze, 5.1° on UT Multiview	–	No	MPIIGaze, UT Multiview
	Yu et al. [39]	2018	5.4° on Eyediap, 5.7° on UT Multiview	–	No	UTMultiview, Eyediap

3 Discussions and Conclusion

3.1 Discussions

As shown in Table 1, each type of method has its advantages and limitations. It is difficult to compare the accuracy accurately because of the different evaluation criteria, but it is obvious that all the methods can achieve a good performance with a high estimation accuracy. However, most of them only can handle small head movements and achieve high accuracy in the special cases. With the development of commercial application, it is an inevitable trend to propose gaze tracking method with free head movement. Apart from the head pose, most methods require calibration procedure. The complex calibration process is inconvenient for the commercial application. To reduce the calibration points or without calibration is the direction of future development. Methods in recent years have tended to study gaze tracking without calibration, but the price is the reduction of tracking accuracy. Therefore, how to maintain a high tracking accuracy without personal calibration is one of the future research directions.

Obviously, appearance-based gaze tracking methods have showed great potential in various applications and achieved a high tracking accuracy, but some challenges still exist and need to be further researched.

(a) Free head movements

Most of the appearance-based gaze tracking methods require the user under a fixed head pose or a limited head moving range, which limit the user's moving space and result in a bad user experience in HCI applications. Therefore, it is necessary to develop an effective gaze tracking method that can handle free head movement. The latest methods take this problem into account, and most of the algorithms add the head pose vector to the feature extraction process. However, experiments show that this method is far from enough to offset the errors caused by head posture.

(b) Non-calibration or auto-calibration

Since the difference of personal eye parameters in different individuals, most gaze tracking methods require personal calibration process. On the one hand, calibration

process requires user's involvement, which leads to a low degree of automation. On the other hand, calibration accuracy greatly affects the gaze tracking accuracy. Therefore, it is urgent to propose a robust gaze tracking method with auto-calibration or without calibration to achieve automation and stability. Although the latest algorithms almost reduce the calibration process, some experiments show that individual calibration in the testing process still has an optimization effect on the accuracy of the algorithm. Therefore, to solve this problem, the calibration process should be avoided in the training process and achieve robustness and automation, while the testing process can use the calibration appropriately, as long as the calibration process is not complicated.

(c) Reduction of training samples

Appearance-based gaze tracking methods normally require large training samples to obtain high accuracy. However, this will bring large computational cost. KNN, SVM and GP algorithms do not support large data sets, and neural network method also takes a long time. Therefore, how to reduce the training samples by discovering some more effective feature descriptors to improve the tracking efficiency while maintaining the accuracy remains a challenging task.

(d) Incorporating the merits of multiple methods

In recent years, all kinds of methods have their advantages and disadvantages. Taking the advantages of good methods and avoiding the disadvantages can greatly improve the calculation cost and accuracy of methods. For example, Wang et al. [18] uses the advantages of convolutional neural network to extract depth features, and uses the advantages of parallel computation of random forests to accelerate the mapping speed.

3.2 Conclusion

In this paper, a review on appearance-based gaze tracking methods has been presented. The mainstream feature input extraction methods based on human-eye appearance image have been introduced. Five classified appearance-based gaze tracking methods according to the mapping manner have been presented in detail. Finally, four challenging issues have been summarized and discussed for future research.

Acknowledgement. This work was supported by National Natural Science Foundation of China (61876168, U1509207), National Key R&D Program of China (2018YFB1305200), and Zhejiang Provincial Natural Science Foundation of China (LY18F030020).

References

1. Stiefelhagen, R., Yang, J.: Gaze tracking for multimodal human-computer interaction. In: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, Munich (1997)
2. Morimoto, C.H., Mimica, M.R.M.: Eye gaze tracking techniques for interactive applications. Comput. Vis. Image Underst. **98**(1), 4–24 (2015)
3. Sawahata, Y., Khosla, R., Komine, K.: Determining comprehension and quality of TV programs using eye-gaze tracking. Pattern Recogn. **41**(5), 1610–1626 (2008)
4. Guan, Q., Tang, F., Zhou, X., Min, H.: A survey of 3D eye model based gaze tracking. J. Comput.-Aided Des. Comput. Graph. **29**(9), 1579–1589 (2017)

5. Pham, C., Thiele, S., Parkinson, J., Li, S.: Alcohol warning label awareness and attention: a multi-method study. Alcohol Alcohol. **53**(1), 1–7 (2017)
6. Pfeiffer, T.: Towards gaze interaction in immersive virtual reality : evaluation of a monocular eye tracking set-up. In: Schumann, M., Kuhlen, T. (eds.) Virtuelle und Erweiterte RealitatFunfter Workshop der GIFachgruppe VRAR, pp. 81–92. Shaker Verlag (2008)
7. Valenti, R., Gevers, T.: Accurate eye center location and tracking using isophote curvature. In: IEEE Conference on Computer Vision & Pattern Recognition, vol. 1, pp. 1–8. IEEE, Alaska (2008)
8. Valenti, R., Staiano, J., Sebe, N., Gevers, T.: Webcam-based visual gaze estimation. In: Foggia, P., Sansone, C., Vento, M. (eds.) ICIAP 2009. LNCS, vol. 5716, pp. 662–671. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04146-4_71
9. Guestrin, E.D., Eizenman, M.: General theory of remote gaze estimation using the pupil center and corneal reflections. IEEE Trans. Biomed. Eng. **53**(6), 1124–1133 (2006)
10. Zhang, Y., Bulling, A., Gellersen, H.: Discrimination of gaze directions using low-level eye image features. In: Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-Based Interaction, pp. 9–14. ACM, Beijing (2011)
11. Wang, Y.: Appearance-based gaze estimation using deep features and random forest regression. Knowl.-Based Syst. **110**, 293–301 (2016)
12. Tan, K., Kriegman, D.J., Ahuja, N.: Appearance-based eye gaze estimation. In: 6th IEEE Workshop on Applications of Computer Vision, p. 191. IEEE Computer Society, Orlando (2002)
13. Martinez, F., Carbone, A., Pissaloux, E.: Gaze estimation using local features and non-linear regression. In: 19th IEEE International Conference on Image Processing, pp. 1961–1964. IEEE, Orlando (2013)
14. Guo, Z., Zhou, Z., Liu, Z.: Appearance-based gaze estimation under slight head motion. Multimed. Tools Appl. **76**(2), 2203–2222 (2016)
15. Wood, E.,Tadas, B., Louis, M., Peter, R., Andreas, B.: Learning an appearance-based gaze estimator from one million synthesised images. In: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and Applications, pp. 131–138. ACM, New York (2016)
16. Zhang,Y., Bulling, A., Gellersen, H.: Discrimination of gaze directions using low-level eye image features. In: Proceedings of the 1st International Workshop on Pervasive Eye Tracking & Mobile Eye-based Interaction, pp. 9–14. ACM, New York (2011)
17. Wang, Y., Zhao, T., Ding, X.: Learning a gaze estimator with neighbor selection from large-scale synthetic eye images. Knowl.-Based Syst. **139**, 41–49 (2017)
18. Wang, Y.: Appearance-based gaze estimation using deep features and random forest regression. Knowl. Based Syst. **110**, 293–301 (2016)
19. Sugano, Y., Matsushita, Y., Sato, Y.: Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1821–1828. IEEE, Columbus (2014)
20. Kacete, A., Séguier, R., Collobert, M., Royan, J.: Unconstrained gaze estimation using random forest regression voting. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (eds.) ACCV 2016. LNCS, vol. 10113, pp. 419–432. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54187-7_28
21. Huang, Q., Veeraraghavan, A., Sabharwal, A.: Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. Mach. Vis. Appl. **28**(5–6), 445–461 (2017)
22. Ounpraseuth, S.T.: Gaussian processes for machine learning. Int. J. Neural Syst. **14**(2), 69–106 (2004)

23. Wojke, N.: Gaze-estimation for consumer-grade cameras using a Gaussian process latent variable model. Pattern Recogn. Image Anal. **26**(1), 248–255 (2016)
24. Williams, O., Blake, A.: Sparse and semi-supervised visual mapping with the S^3 GP. In: 2016 IEEE Computer Society Conference on Computer Vision & Pattern Recognition, pp. 230–237. IEEE, New York (2006)
25. Ferhat, O., Vilarino, F., Sánchez, F.J.: A cheap portable eye-tracker solution for common setups. J. Eye Mov. Res. **7**(3), 1–10 (2014)
26. Sugano, Y., Matsushita, Y.: Appearance-based gaze estimation using visual saliency. IEEE Trans. Pattern Anal. Mach. Intell. **35**(2), 329–341 (2013)
27. Huang, Y., Dong, X., Hao, M.: Eye gaze calibration based on support vector regression machine. In: 9th World Congress on Intelligent Control and Automation, pp. 454–456. IEEE, Taipei (2011)
28. Zhu, Z., Ji, Q., Bennett, K.P.: Nonlinear eye gaze mapping function estimation via support vector regression. In: 18th International Conference on Pattern Recognition, pp. 1132–1135. IEEE, Hong Kong (2006)
29. Wu, Y.L., Yeh, C.T., Wei, H.: Gaze direction estimation using support vector machine with active appearance model. Multimed. Tools Appl. **70**(3), 2037–2062 (2014)
30. Chuang, M.-C., Bala, R., Bernal, E.A., Paul, P., Burry, A.: Estimating gaze direction of vehicle drivers using a smartphone camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 165–170. IEEE, Columbus (2014)
31. Baluja, S., Pomerleau, D.: Non-intrusive gaze tracking using artificial neural networks. Adv. Neural. Inf. Process. Syst. **98**(1), 753–760 (1993)
32. Yu, L., Xu, J., Huang, S.: Eye-gaze tracking system based on particle swarm optimization and BP neural network. In: 12th World Congress on Intelligent Control and Automation, pp. 1269–1273. IEEE, Guilin (2016)
33. Sewell, W., Komogortsev, O.: Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. In: Mynatt, E.D., Schoner, D., Fitzpatrick, G., Hudson, S.E., Edwards, W.K., Rodden, T. (eds.) Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI EA 2010), pp. 3739–3744. ACM, Atlanta (2010)
34. Zhang, X., Sugano, Y., Fritz, M.: It's written all over your face: full-face appearance-based gaze estimation. Comput. Vis. Pattern Recogn. **1**(5), 2299–2308 (2016)
35. Cheng, Y., Lu, F., Zhang, X.: Appearance-based gaze estimation via evaluation-guided asymmetric regression. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 105–121. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_7
36. Zhang, C., Rui, Y., Cai, J.: Efficient eye typing with 9-direction gaze estimation. Multimed. Tools Appl. **77**(15), 1–18 (2017)
37. Palmero, C., Selva, J., Bagheri, M. A., Escalera, S.: Recurrent CNN for 3D gaze estimation using appearance and shape cues. Comput. Vis. Pattern Recogn. **1**(3), 1–13 (2018)
38. Fischer, T., Chang, H.J., Demiris, Y.: RT-GENE: real-time eye gaze estimation in natural environments. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 339–357. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_21
39. Yu, Y., Liu, G., Odobez, J.-M.: Deep multitask gaze estimation with a constrained landmark-gaze model. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11130, pp. 456–474. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11012-3_35