Estimating Point of Regard with a consumer camera at a distance

Jordi Mansanet[1] [⋆], Alberto Albiol[1], Roberto Paredes[2], Jose Manuel Mossi[1], and Antonio Albiol[1]

[1] iTEAM - Instituto de Telecomunicaciones y Aplicaciones Multimedia
alalbiol@iteam.upv.es
[2] ITI - Insitituto Tecnológico de Informática
rparedes@dsic.upv.es
Universitat Politècnica de València
Valencia Spain

Abstract. In this work, we have studied the viability of a novel technique to estimate the POR that only requires video feed from a consumer camera. The system can work under uncontrolled light conditions and does not require any complex hardware setup. To that end we propose a system that uses PCA feature extraction from the eyes region followed by non-linear regression. We evaluated three state of the art non-linear regression algorithms. In the study, we also compared the performance using a high quality webcam versus a Kinect sensor. We found, that despite the relatively low quality of the Kinect images it achieves similar performance compared to the high quality camera. These results show that the proposed approach could be extended to estimate POR in a completely non-intrusive way.

Keywords: point-of-regard, human computer interaction (HCI), gaze estimation, eye tracking

1 Introduction and related works

The problem of gaze estimation and tracking has attracted increasing attention recently because of its many potential applications. Since eye gaze provides relevant information about the user's attention and intention, the applications of eye gaze trackers are usually categorized into diagnostic or interactive [11]. A few examples of diagnostic applications include ophthalmology, neurology, psychology and more recently advertisement and marketing. In the context of interactive applications eye gaze can be used for eye typing, computer control, and gaming among many others.

Gaze tracking, also commonly referred to as Point of Regard (POR) estimation, is used to identify the exact point where the person is looking at. The

⋆ The work presented in this paper has been funded by the Spanish Ministry of Education under the CICYT contract TEVISMART, TEC2009-09146.

two components that determine the POR are the head pose and the relative orientation between the eyes and the head.

The problem of POR estimation still remains a challenge due to some inherent problems such as occlusion, variability between different people's eyes, lighting conditions, variation in scale, etc. In addition, other related problems such as face tracking, eye tracking and head's pose estimation need also to be addressed.

There are many techniques in the literature to estimate the POR [6]. Many of them are quite intrusive, using infrared illumination or complex and expensive hardware systems [12] [16]. Also, some of them require a controlled environment to work properly. Although these techniques achieve high precision, there are some kind of applications where other aspects, such as the intrusiveness, the illumination requirements, the complexity of the hardware, the cost, etc. are more important at the expense of lower precision.

Most of the POR estimation techniques are based on extracting feature points on the eyes, like the pupil or the glint (using active light). Once the feature points are detected, the techniques are grouped into *model* and *regression-based* methods. In the first case, gaze direction is obtained using a geometric model of the eye [17] [13]. On the other hand, interpolation-based methods assume that the mapping from eye feature points to gaze coordinates can be estimated by regression techniques. A few examples of regression techniques used to estimate the POR are the polynomial parametric model [11] and the neural network model proposed in [7].

Appearance-based POR estimation techniques do not detect any feature points on the eyes. Instead, they build a function that maps the eye image pixels to the POR directly. Baluja and Pomerleau [1] use multilayer network to estimate the regression. Tan et al. [15] employ Local Linear Embedding to learn the eye image manifold. Williams et al. [18] use a sparse Gaussian process interpolation method on filtered visible spectrum images. Typically, these methods do not require camera calibration since the mapping is made directly from the image pixels. Also, the error is minimized because is not necessary to locate specific features on the eyes.

In this work, we have studied the viability of a novel appearance-based POR estimation technique using a low cost consumer camera at a distance. Our approach does not require any particular lighting conditions nor expensive hardware. As mentioned above, POR estimation depends on both head pose and eyes orientation. In this paper we only focus on the second component, i.e. the eyes component. For this reason, in our experiments we used a metallic structure to fix the user's head to it. In our study we also want to know the effect of the image quality on the POR estimation. For this reason a comparison between different consumer cameras is presented.

The rest of the paper is organized as follows. A system overview is described in section 2. Section 3 details the databases created with the different consumer cameras compared. In section 4, we explain the non linear regression techniques employed and the feature extractor process. The results of the study are presented in section 5, and we make a conclusion and discuss the future work in section 6.

2 System overview

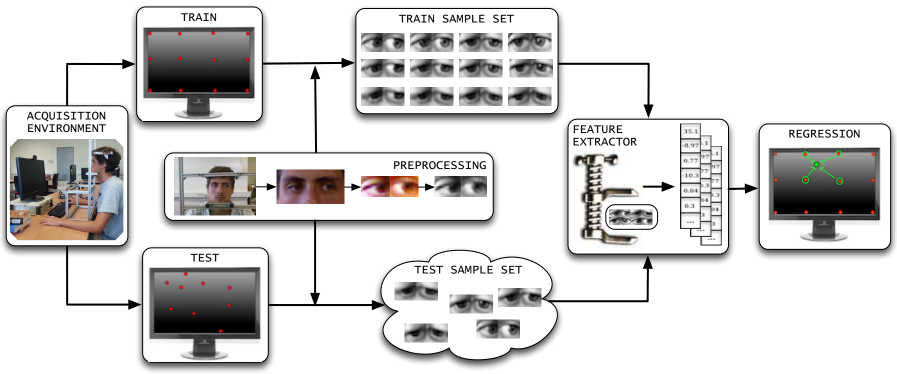


Fig. 1: Flowchart of proposed approach.

Figure 1 shows the general flowchart of the proposed approach. In the first stage, the person is situated in front of the screen and his head is fixed to a metallic structure to avoid pose changes. To generate training, validation and test data, the user is requested to look at different points of the screen. During the training, the screen points are distributed on a regular grid. We selected a 4×3 grid which we found was a good compromise to avoid a long and tedious calibration process and obtain a representative training sample set, as shown in Fig. 1. For the validation and test data, the points are randomly located on the screen to obtain more realistic error measures.

Once the images are obtained, eyes are located and cropped. To do this, we used the software developed by Jason M. Saragih [14]. Finally, the eyes images are transformed to grayscale and normalized to zero mean and unit variance.

Since the image pixels are highly correlated, it's convenient to use dimensionality reduction techniques to extract relevant features, as we will explain later. Using these features, we trained and compared a few regression algorithms to obtain the POR estimation. Because this estimation may be noisy, we could use a Kalman filter [8] that smooths the POR values along time. It is important to mention that the models are trained independently for each person of the database.

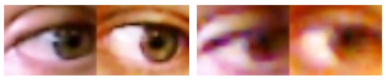

(a) Logitech (b) Kinect

Fig. 2: Example of person's eye obtained with both consumer cameras.

3 Database

To study the viability of the algorithms employed, we have created two databases using two different consumer cameras. The first one was the Logitech webcam HD C525 [3], which provides color images at a resolution of 1280×720 . The second camera was the Microsoft Kinect sensor [9], which additionally to depth information provides 640×480 RGB images. In Figure 2, we can compare the quality of the images obtained with both cameras. Although it is evident the better quality of the Logitech images, the Kinect sensor has the advantage of providing depth information, which can be very useful to compensate head pose.

In our experiments, we collected images from seven different people without any particular restriction (persons are allowed to wear glasses and any gender and age). As we mentioned above, during the training the user is requested to look at 12 positions located on a grid over the screen (see Fig. 1 for an example of the eyes images obtained from each position). For the validation and test, each user was requested to look at 10 random positions on the screen. It is worth to mention, that in our experiments we used a 20 inches screen with a 1680x1050 resolution, and the face is situated about $50cm$ from the screen. This information is important because in the results section we evaluate the error as a pixel distance to the ground truth.

In the Table 1, we summarize the specifications of the databases.

	number of persons	images per person	images per position	number of positions	distribution points
train	7	360	30	12	grid
validation	7	150	15	10	random
test	7	150	15	10	random

Table 1: Databases specifications.

4 Feature extractor and non linear regression

After normalization, we obtain 82×31 cropped eyes images, what yields a total of 2.542 pixel features. To reduce the dimensionality of this feature vector we used Principal Component Analysis (PCA) [10]. The number of principal components is chosen to keep the 95% of the energy of the training samples, what allows to obtain 18 PCA features.

Using the PCA features, we estimate the POR value using a non linear regression technique. We have compared three state of the art methods: k-Nearest Neighbor Regression (kNNR), Support Vector Regression (ϵ-SVR) [5] and Random Forest Regression (RFR) [2].

In kNNR, the values of the k nearest neighbors are averaged to get the estimation. To increase the performance, the contribution of each neighbor is weighted according to its distance. In particular, a weight of $1/d^2$ is used, where d is the distance to the neighbor. We evaluated three common distances: L0, L1 and L2. The other parameter that was optimized was the number of neighbors (k).

ϵ-SVR [5] is a supervised algorithm based on SVM. To perform non linear regression we compared three different kernels: polynomial, radial basis function and sigmoid. For each kernel the parameter C was optimized.

RFR [2] is a regression algorithm that uses a set of regression trees. We have compared two types of weak binary classifiers. The first one, which we call *original*, compares a random PCA feature with a threshold. The second one, (dif), uses the difference between two random PCA features. Finally, the output of the regression forest is obtained by averaging the prediction of each single tree. For this reason, an important parameter to optimize is the number of trees of the forest (nt).

5 Results

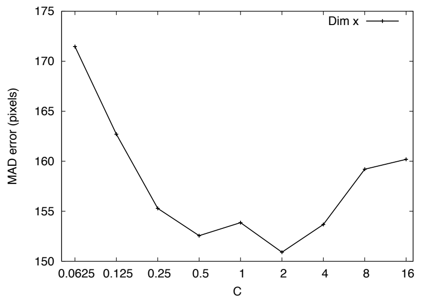
In our experiments, we used training and validation data to train and tune the parameters of the different regression algorithms. To measure the POR performance for a person of the database, we used the Mean Absolute Deviation (MAD), which can be computed as:

$$MAD = \frac{1}{N}\sum_{n=1}^{N} |f_{\boldsymbol{x}_n} - f_\theta(\boldsymbol{x}_n)| \tag{1}$$

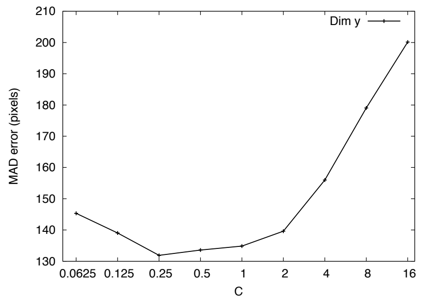
where N is the number of samples, $f_{\boldsymbol{x}_n}$ is the ground truth, and $f_\theta(\boldsymbol{x}_n)$ is the value estimated by the regression function. The overall system performance is obtained by averaging the MAD results of all the people in the database.

Figure 3 displays the results of the parameter optimization process for all the proposed regression algorithms. Note that the ϵ-SVR implementation used in this work (LIBSVM [4]) optimizes the gaze position for each spatial dimension independently.

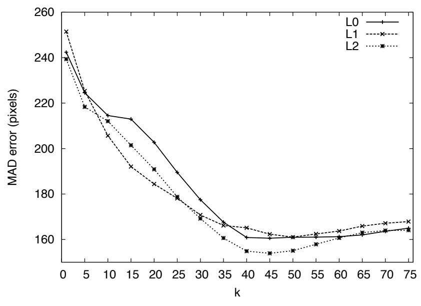
After parameter tuning using the validation data, we choose the following configuration for each algorithm that provides the best performance. In the case of ϵ-SVR we selected a polynomial kernel with $C_x = 2$ and $C_y = 0.25$ for each dimension, respectively. For kNNR we choose $k = 45$ with the $L2$ distance. Finally, for RFR we have used 100 trees and dif as a weak binary classifier.



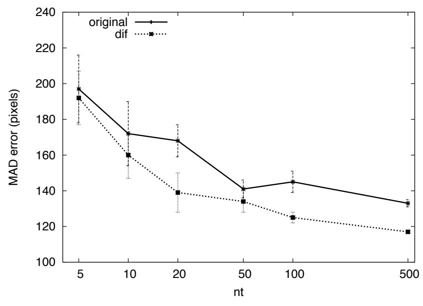
(a) MAD error as a function of the parameter C for x-dimension for the ϵ-SVR algorithm.



(b) MAD error as a function of the parameter C for y-dimension for the ϵ-SVR algorithm.



(c) MAD error as a function of the number of neighbors (k) using different distances (L0, L1 and L2) for the kNNR algorithm.



(d) MAD error as a function of the number of trees (nt), comparing *original* and dif as a weak classifiers, for the RFR algorithm. Each plotted point is an average computed from 4 repetitions of the experiment.

Fig. 3: Results of the training process for each regression algorithm.

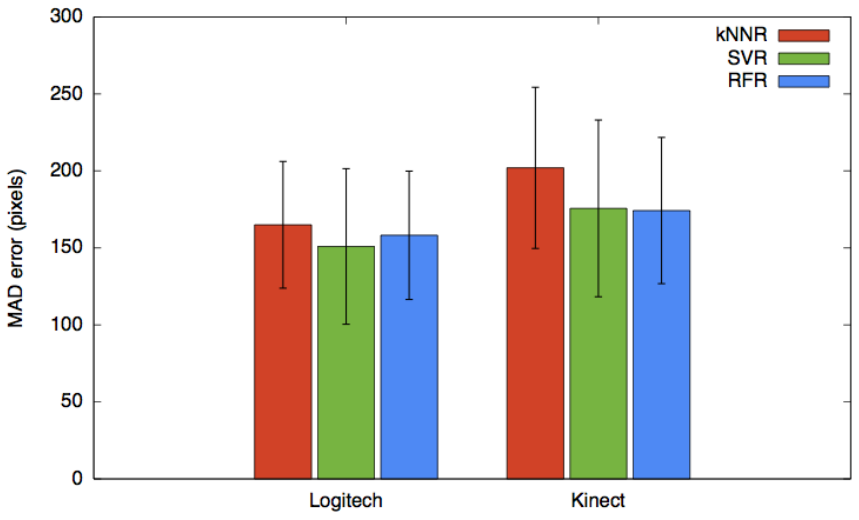


Fig. 4: MAD error bars for each algorithm for both consumer cameras. The result is an average computed from all candidates of the database.

Using the previous configurations, we evaluated the regression algorithms with the test set. The obtained results are displayed in the Fig. 4 for each consumer camera. As we can see in the bar chart, the Logitech camera performs slightly better than the Kinect, as it could be expected due to its better image quality. Regarding the regression algorithms, ϵ-SVR and RFR obtain very similar results. On the other hand, kNNR works worse probably because the regular grid to map the regression space is not dense enough. In any case the smallest achieved MAD error is around 150 pixels. Considering our screen size, the proposed system is able to distinguish around 10 horizontal and 7 vertical zones on the screen, which is more than enough for many applications.

6 Conclusions and future work

In this work we have shown the viability of POR estimation using a consumer camera at a distance. The proposed system does not require any active source of illumination and can adapt to any person after a simple personal calibration process. The evaluation results show that the system achieves a relatively high accuracy even if just a few points are used to train the regression algorithms.

We also found that the Kinect sensor performance is not much worse than the Logitech webcam, despite the poor quality of its RGB images. This opens the door to the possibility of using Kinect depth information to correct head's pose variations.

It is worth to mention that we have built a database containing images of the Logitech camera and the Kinect sensor from seven different people. This database allows us to train and test the regression algorithms to estimate the POR, and to compare the performance of the system using both consumer cameras.

Once the viability of the technique has been proved, the next step would be to remove the limitation of head's fixation by coupling head pose estimation. Also different dimensionality reduction techniques could be explored, in particular algorithms that preserve the topology of the original space are very promising in this context.

References

1. S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical report, 1994.
2. L. Breiman. Random forests. *Machine Learning*, 2001.
3. Logitech HD Webcam C525. `http://www.logitech.com/es-es/webcam-communications/webcams/hd-webcam-c525`.
4. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
5. H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines, 1996.
6. D.W. Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *PAMI, IEEE Transactions*, 2010.
7. Q. Ji and X. Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 2002.
8. R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 1960.
9. Microsoft Kinect. `http://www.microsoft.com/en-us/kinectforwindows`.
10. Timmerman M.E. Principal component analysis (2nd ed.). i. t. jolliffe. *Journal of the American Statistical Association*, 2003.
11. C.H. Morimoto and M.R.M. Mimica. Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.*, 2005.
12. F. Pirri, M. Pizzoli, and A. Rudi. A general method for the point of regard estimation in 3d space. In *Proceedings of the IEEE Conference on CVPR*, 2011.
13. M.J. Reale, S. Canavan, L. Yin, K. Hu, and T. Hung. A multi-gesture interaction system using a 3-d iris disk model for gaze estimation and an active appearance model for 3-d hand pointing. *IEEE Transactions on Multimedia*, 2011.
14. J.M. Saragih, S. Lucey, and J.F. Cohn. Face alignment through subspace constrained mean-shifts. In *International Conference of Computer Vision (ICCV)*, 2009.
15. Kar-Han T., D.J. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In *Applications of Computer Vision*, 2002.
16. K. Takemura, Y. Kohashi, T. Suenaga, J. Takamatsu, and T. Ogasawara. Estimating 3d point-of-regard and visualizing gaze trajectories under natural head movements. In *Symposium on Eye-Tracking Research and Applications*, 2010.
17. A. Villanueva, R. Cabeza, and S. Porta. Eye tracking: Pupil orientation geometrical modeling. *Image and Vision Computing*, 2006.
18. O. Williams, A. Blake, and R. Cipolla. Sparse and semi-supervised visual mapping with the s3gp. In *IEEE Computer Society Conference on CVPR*, 2006.