Indirect Eye Gaze Estimation based on Depth Information

Tomas S. Correa, Mario F. M. Campos, William Robson Schwartz, Erickson R. Nascimento
Universidade Federal de Minas Gerais
Department of Computer Science
Belo Horizonte, Minas Gerais, Brazil
Email: {tomassc, mario, william, erickson}@dcc.ufmg.br

***Abstract*—Eye gaze estimation has been exploited over the years to enable interaction between the user and a computer system through eye and head movements. In addition, there are applications requiring the estimation of regions in which the user is more interested, such as placement of user-specific advertisement. There are two main approaches to estimate the eye gaze, one is based on intrusive eye gaze trackers and the other is based on data acquired by cameras. Even thought the former approach provides a higher accuracy than the latter, it requires the attachment of devices to the user, which might restrict user's movements. This work proposes a method to estimate the eye gaze in an indirect manner, in which a depth sensor is imaging the user's face that is looking at a wall being captured by a second camera. The goal is to estimate the eye gaze of the user in the image captured by the second camera. Experimental results demonstrate the viability of the proposed indirect eye gaze estimation method.**

***Keywords*-eye gaze estimation; depth information; head pose estimation;**

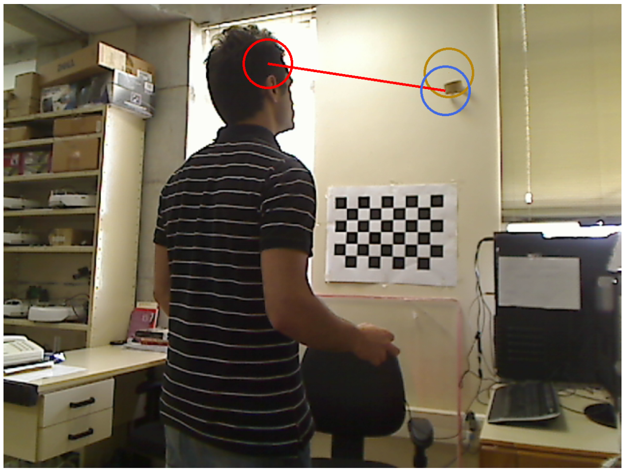


Fig. 1. Example of an user looking at the wall. The blue circle determines the groundtruth position of the region and the yellow circle represents the position estimated by our system.

I. Introduction

Information regarding the eye gaze has been exploited in the literature for several years [1]. The main motivation for that is to enable interactions between an user and computer systems in a very subtle and non-intrusive manner based only on the eyes and head movements. Besides interaction with computer systems [2], other relevant applications that might benefit from eye gaze estimation are the location of regions of interest in a wall displaying some information and person-specific advertisement based on the persistent interest of the user in a specific product capture by his/her eye gaze.

Research on eye gaze aims at estimating the direction a person is gazing. This field has been studied for several years focusing on two main approaches: intrusive eye gaze trackers and camera-based eye gaze trackers [1]. While the former approach uses extra devices such as contact lenses [3] or measurements of skin potentials [4], the latter considers only information of the eye that can be captured by a camera, which is less accurate but does not require special equipment and is non-intrusive, which is highly desirable. This work focuses on the latter approach.

Among the approaches based on cameras, several techniques have been employed, Stiefelhagen et al. [5] employed a neural network trained to estimate the eye orientation while maintaining the head fixed, which limits the user's movements. Recently, Zhang et al. [2] proposed a gaze interface that does not require the head to be fixed to allow the interaction between the user and a computer system through the computer screen focusing on possible advertisement applications. Other works have been proposed in the field of head pose estimation to support the eye gaze estimation, such as [6] which uses depth information and several other approaches that can be found in [7].

Differently from previously proposed works, which aim at estimating the eye gaze to interact to the computer system through a screen, this paper proposes an approach to estimate the indirect eye gaze, in which a depth sensor is imaging the user's face that is looking at a wall being captured by a second camera. Our goal is to estimate the eye gaze of the person in the image captured by the second camera (image of the wall). Figure 1 shows an example of the second image and a mark in the wall. The estimation of the eye gaze in this image is made using the depth sensor placed in the user's right-hand side.

The proposed approach is composed of two main steps. First, the head pose estimation is performed based on the depth information captured by the Kinect camera [8]. With that information, we are able to estimate the angle between the

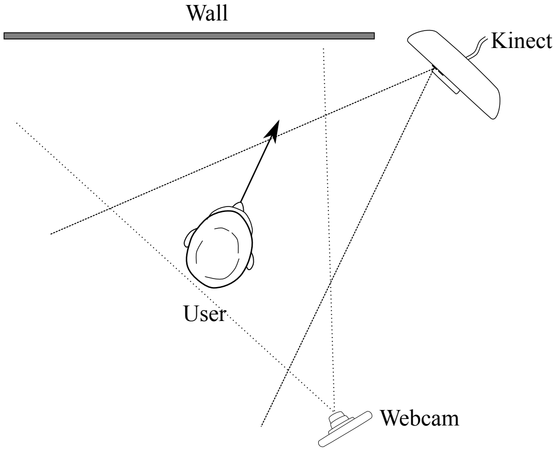


Fig. 2. Diagram of our experimental setup.

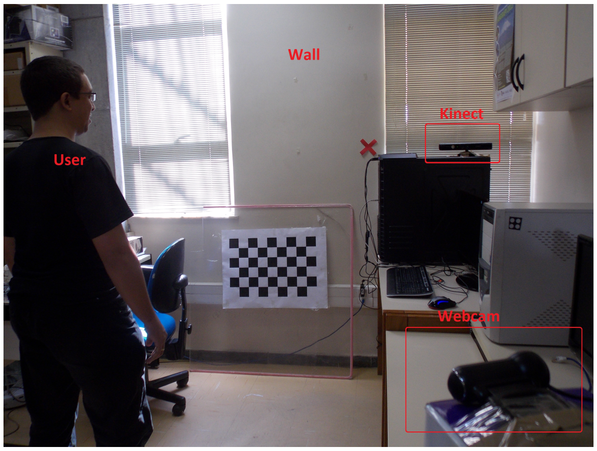


Fig. 3. Configuration of the experimental setup. The experiments consisted of users positioned 2 meters away from a wall looking at a red X.

user's face and the wall he/his is looking at and the location in which the "eye gaze" will intersect the wall. Then, the second stage locates in the image of the wall captured by the second camera, the region were the intersection with the wall will take place (region that the user is looking at).

The remaining of the paper is divided as follows. Section II describes the methodology developed to perform indirect eye gaze estimation. Then, an experimental validation and a discussion regarding the difficulties found are presented in Section III. Finally, Section IV concludes this work with final remarks and directions to further developments for the method.

II. Methodology

In this section we detail our gaze estimation system. Our system computes the user's gaze using an estimation of his head pose. We consider that the users will be away from the region of interest, thus the head orientation can provide a closer approximation of the point of regard.

Our system requires a setup composed of a RGB-D sensor and a Webcam. The RGB-D sensor is used for head pose estimation and the Webcam provides a visualization of the region where the user is looking. Figure 2 shows a diagram with the position of each device in our system. A picture of the prototype is showed in Figure 3.

The system has two phases: the calibration and the execution phase. In the first phase, for setting up the system, a RGB-D sensor (in this work we used a Kinect camera) and a Webcam are calibrated by using the Camera Calibration Toolbox of the Matlab [9].

The intrinsic calibration of both Kinect camera and Webcam are estimated as well as the extrinsic calibration of the stereo system composed of the two cameras. These calibrations provide the focal distance of Kinect and Webcam (f_k and f_w respectively) and the transformation between the Kinect and Webcam coordinate system (rotation matrix R and translational vector T). Thus, our system is capable of determining the position of Webcam relative to the Kinect RGB camera and vice versa, allowing the transformation from one coordinate system to the other. A third calibration was performed to determine the location of the wall in space. This location is given based on the Webcam's coordinate system.

In execution phase, the main modules of the system are executed. Our system is composed of two modules: the head pose estimation and a second module where is computed the region of interest of users. The system's workflow can be divided into 5 steps:

1) The persons pose $\mathbf{P}$ of the head is detected using a Kinect camera;
2) This pose $\mathbf{P}$ is used to create a view direction vector $\mathbf{V}$ in Kinects coordinate system;
3) The vector $\mathbf{V}$ is transformed to Webcams coordinate system using the extrinsincs parameters R (rotation matrix) and T (translational vector). The final vector is computed as $\mathbf{V}_w = R * \mathbf{V} + T$. Also, we transform the point $\mathbf{H}$, which is the location of the user's head.
4) We compute the intersection point $\mathbf{I} \in R^3$ between the vector $\mathbf{V}$, starting at $\mathbf{H}$, and the pre-calculated plane of the wall;
5) This 3D point (which is in the coordinate system of the camera) is mapped to the corresponding 2D point in the image.

A. Head Pose Estimation

The goal of this module is to estimate the heads pose of a user. In order to compute the orientation of user's head, we use the estimator presented in Microsoft Face Tracking Software Development Kit (MFTSD) [10]. This Microsoft estimator uses input data from Kinect [8] and provides some information, such as head pose and facial expressions, in real time.

The head pose $\mathbf{P}$ is computed in a 6-dimensional space and it's given relative to Kinect position. In every frame, the MFTSD framework outputs X, Y, and Z position of the users head (point $\mathbf{H}$) and the angles for pitch, roll and yaw, which describe the head's orientation. By using the head location (X, Y, Z) and orientation, we are able to determine the vector

Target	X error	Y error	Distance error
T1	19.61	50.67	56.16
T2	19.64	32.58	34.18
T3	47.29	41.89	56.26
T4	43.15	47.01	60.48
Average	32.42	43.04	51.77
STD	12.88	6.80	10.30

TABLE I
ROOT-MEAN-SQUARE ERROR (RMSE) IN X AND Y COORDINATES AND EUCLIDIAN DISTANCE. THE AVERAGE AND STANDARD DEVIATION OF EACH POINT IS ALSO SHOWN. THE VALUES ARE GIVEN IN PIXELS.

V. This vector represents the direction which user is staring at. Figure 1 depicts this vector as a red line.

B. Computing the Region of Interest

By using the plane of the wall in 3D space, in the format $ax+by+cz+d=0$, and the parametric equations of the line determined by the point **H** and the vector $\mathbf{V}_w = R * \mathbf{V} + T$, we calculate the intersection point **I** between this line and the plane.

The point I is represented in 3D space and determines a point in space in Webcams view. With correspondence formulas and using intrinsic parameters of the Webcam, we determine the 2D point on image corresponding to that 3D point as:

$$x = \frac{f_w * I_x}{I_z} + c_x; \quad y = \frac{f_w * I_y}{I_z} + c_y, \qquad (1)$$

where x and y are the final coordinates in 2D space, I_x, I_y and I_z are the 3D coordinates of the intersection point **I**, f_w, f_w is the focal distance, and c_x and c_y are center of image for each axis (intrinsic parameters of the Webcam).

III. EXPERIMENTS

In order to evaluate the accuracy of our system, we performed several experiments with 11 users staring at four targets separated by 400 mm (about 86 pixels) in a wall. Figure 8 (a) shows these four targets. They are represented by red X signals.

In our experiments, the users were about 2 meters away from the wall and they looked at each target three times (Figure 3). Considering that the worst measure can be due to human error (they did not point the nose to the targets as advised), we removed the worst result among the three estimations. Figure 7 shows the best estimations provided by our system for each target.

The final result was computed as average of the two estimations for each user. Table I shows the root-mean-square-error (RMSE) of these averages for each one of the four targets. Additionally, the errors for X and Y directions of the wall image are shown. The measures are given in pixels.

The plot in Figure 4 shows the average error and its standard deviation. It is possible to notice that Target 2 has less error than the others.

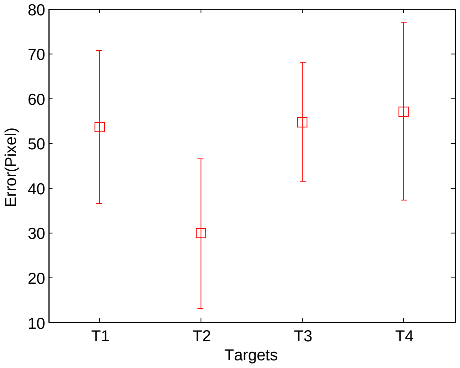


Fig. 4. Average error (in pixels) and standard deviation for all targets.

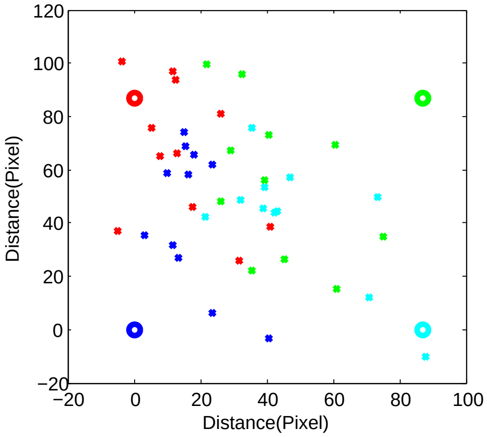


Fig. 5. The 44 target locations estimated in the experiments are represented by color crosses. The circles show the real location for each target. The color of a cross is related to its corresponding circle.

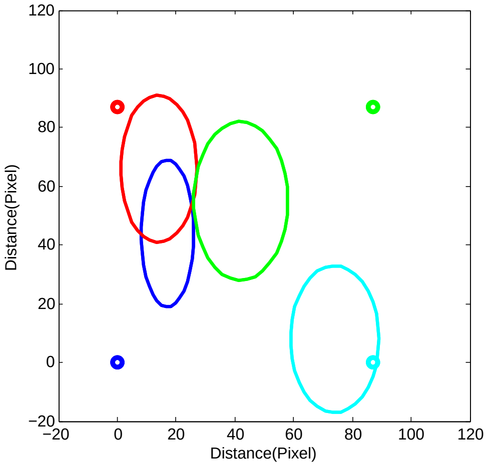
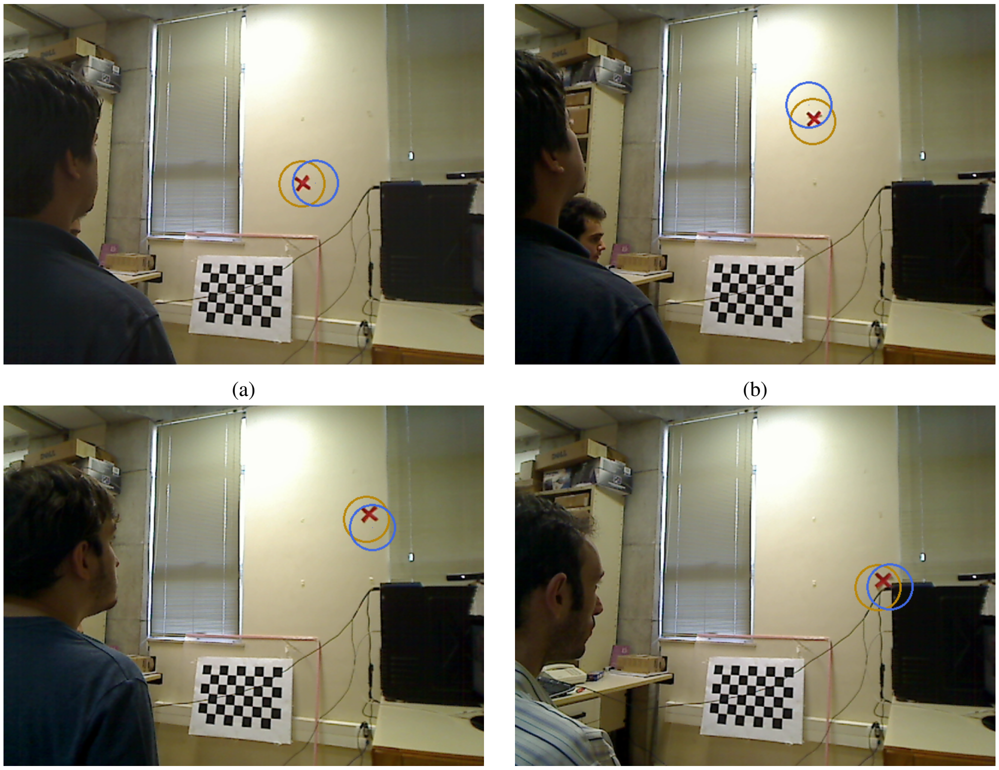


Fig. 6. Average error and its standard deviation for all four targets. The ellipses are related to a target by their color.

The average and standard deviation of the errors for each target considering all users are shown in Figure 6. We can note that there are a clear bias for the error in Targets 1 (blue circle) and 3 (green circle). The center of these ellipses are shifted to the center of the image. One of the reason for such results can be a calibration error.



(a) (b)

(c) (d)

Fig. 7. Examples of the gaze estimation for (a) Target 1; (b) Target 2; (c) Target 3 and (d) Target 4. These were the best estimation results of the system.

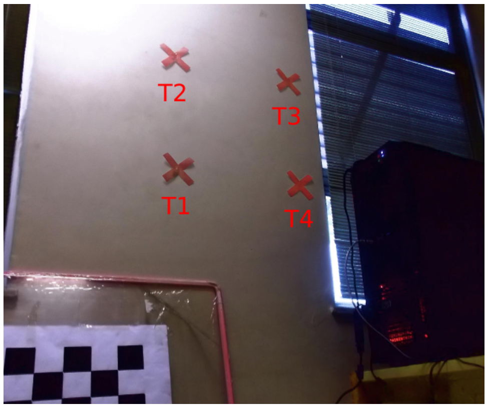


Fig. 8. The four red X represents the targets used in experiments. They are separated by 400 mm (about 86 pixels).

IV. Conclusion and Future Works

In this work we propose a gaze estimation system based on the head pose. There are a large number of applications that will be able to use this work, such as security vigilance, marketing, entertainment (i.e. games) and others.

In spite of the errors presented in experiments section, our system can provide an initial estimation for the user's gaze. In a further approach, the system's accuracy can be increased by improving the calibration phase in order to reduce the coordinate system transformation error. We will also apply the system to create a heat map to analyse the user's attention.

Acknowledgments

The authors are grateful to FAPEMIG, CAPES and CNPq funding agencies.

References

[1] C. H. Morimoto and M. R. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 4–24, 2005.

[2] Y. Zhang, A. Bulling, and H. Gellersen, "Sideways: A gaze interface for spontaneous interaction with situated displays," in *Proc. of the 31st SIGCHI International Conference on Human Factors in Computing Systems*, 2013.

[3] D. A. Robinson, "A method of measuring eye movements using a scleral search coin in a magnetic field," *IEEE Trans. Biomed. Eng.*, vol. 10, pp. 137–145, 1963.

[4] A. Kaufman, A. Bandopadhay, and B. Shaviv, "An eye tracking computer user interface," in *In Proc. of the IEEE Research Frontier in Virtual Reality Workshop*, 1993, pp. 78–84.

[5] R. Stiefelhagen, J. Yang, and A. Waibel, "Tracking eyes and monitoring eye gaze," in *In Workshop on Perceptual User Interfaces, Ban*, 1997.

[6] G. Fanelli, J. Gall, and L. J. V. Gool, "Real time head pose estimation with random regression forests," in *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE, 2011, pp. 617–624.

[7] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, 2009.

[8] Microsoft, "Microsoft kinect," February 2011.

[9] J.-Y. Bouguet, "Camera calibration toolbox for matlab," May 2013, http://www.vision.caltech.edu/bouguetj/calib_doc/.

[10] Microsoft, "Face tracking," May 2013, http://msdn.microsoft.com/en-us/library/jj130970.aspx.