Real-Time Gaze Estimation Using a Kinect and a HD Webcam

Yingbo Li, David S. Monaghan, and Noel E. O'Connor

CLARITY: Center for Sensor Web Technology, Dublin City University, Ireland

Abstract. In human-computer interaction, gaze orientation is an important and promising source of information to demonstrate the attention and focus of users. Gaze detection can also be an extremely useful metric for analysing human mood and affect. Furthermore, gaze can be used as an input method for human-computer interaction. However, currently real-time and accurate gaze estimation is still an open problem. In this paper, we propose a simple and novel estimation model of the real-time gaze direction of a user on a computer screen. This method utilises cheap capturing devices, a HD webcam and a Microsoft Kinect. We consider that the gaze motion from a user facing forwards is composed of the local gaze motion shifted by eye motion and the global gaze motion driven by face motion. We validate our proposed model of gaze estimation and provide experimental evaluation of the reliability and the precision of the method.

Keywords: Gaze estimation, Gaze tracking, Eye tracking, Kinect.

1 Introduction

Novel forms of human-computer interaction are becoming more and more essential in our increasingly highly computerised modern world. This is also evident by an increasing volume of research being carried out in the field of eye and gaze tracking, for example the newly emerging Google Glasses. Human eye and gaze direction can reflect the natural and stable attention of end users and even give insights into mood and human affect. By performing real-time and robust gaze tracking we can track the objects of interest to a user and even design customised and interesting content for the user.

Currently the most popular gaze estimation approaches can be classified into two groups [1] [2] [3] [4] [5]: model-based gaze estimation and screen-based gaze estimation. The former dealing with the graphical object and model before rendering, while the latter manipulates the image at the pixel level. In model-based gaze estimation, researchers estimate the gaze by analysing the model of the eye and the model of the gaze. The authors in [6] estimate the gaze by only a single image of the eye through the iris circle. Kim et al. [7] estimates the gaze by exploiting the position of two eyes in relation to a simple 2D mark. In [8] the authors propose a new method for estimating eye/gaze direction based on appearance-manifolds by utilising the linear interpolation among a small subset

C. Gurrin et al. (Eds.): MMM 2014, Part I, LNCS 8325, pp. 506–517, 2014.

of samples to approximate the nearest manifold point. Head pose and facial expression has also be employed to enhance gaze tracking estimation [9] [10]. The low-cost 3D capturing device, the Microsoft Kinect, has also been exploited to estimate the gaze by modelling the head and eye [11].

In comparison to model-based methods, screen-based methods try to calibrate and build the relation between the eye in 3D space and the gaze on the screen. For example, a video-oculographic gaze tracking system is proposed in [17]. The Microsoft Kinect has also been used as a source for screen gaze tracking [13]. Recently it has been demonstrated how the gaze detection of a user can be used to animate the gaze shifts of virtual characters within virtual environments on the screen [14].

In addition to these ongoing research efforts, the open source software for eye and gaze tracking in both remote or eye-wearing modes has become more and more prevalent in recent years. The most popular of these open sources eye trackers include OpenEyes [18], AsTeRICS [19], Opengazer [20], TrackEye [21] and ITU Gaze Tracker [22]. Within the scope of the work presented here we have utilised a customised version of the ITU Gaze Tracker.

2 Motivation

The authors in [12] have proposed a real-time method to robustly detect the eye (pupil) motion by HD webcam and Microsoft Kinect with the assistance of ITU Gaze Tracker [22]. The authors calibrate and identify the matched pupil centres, by SIFT descriptor, between the images from the Kinect and the Webcam after obtaining the pupil centre from ITU Gaze tracker. By utilising real-time information about the pupil motion and the 3D movement of the head, obtained from the Kinect, the authors can accurately animate the eye motion in a virtual avatar. The resultant avatar thus appears to be more realistic with accurately animated eyes.

It reasonably follows, from the successful detection of the user's eye motion, to be able to estimate the gaze direction from the moving eyes, i.e. The exact place where is the person (or avatar) is looking. In [12] the authors accomplish the task of eye tracking for a user positioned in front of a computer screen and then animate an avatar on the screen. In this scenario the gaze is the focused point on the screen. The ITU Gaze Tracker, exploited in [12], was employed to do the gaze tracking on the screen when the users head/face is a fixed position. However, even a slight facial movement can corrupt the performance of the gaze tracking of ITU Gaze Tracker, based on experimental results.

In [13] the authors proposed gaze estimation by using the face information from the Kinect but this method only tracks the gaze caused by head motion and pose, and does not take into account the motion of the pupils. It is often the case when a user is positioned in front of a computer screen that the gaze is often shifted by moving the pupils, as well as the head position.

In this paper we propose a novel approach to gaze estimation for the case when a user is positioned in front of a computer screen by analysing both the movement

of the pupils and the translational movement of the head using a low cost webcam and a Kinect depth sensor. We have not seen the similar contribution for gaze tracking in front of a computer monitor and capturing devices, which achieves the same accurate gaze tracking by the low-cost devices for around 200euro. The proposed system is very practical because it is often for a person to work and play in front of a computer with capturing device in the current world, while it is meaningful to track his gaze on the computer monitor to learn the real-time attention.

3 An Overview of the Proposed Approach

A HD webcam was used in this approach as it was found, through experimental observation, that the RGB camera in the Kinect was of too low quality for our purposes. The webcam and Kinect are arranged in front and underneath the computer screen, as shown in Figure 1. The eye tracking approach of obtaining the pupil position from Kinect frames follows on from previous work [12]. Firstly the factors influencing the gaze position, including the pitch, yaw and raw of the face are analysed.

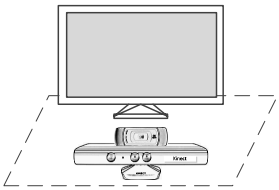


Fig. 1. The designed hardware system

Since the eye and the pupil is on the face, the factors involved in shifting the gaze position can be classified into two groups. The motion of the face, we have termed the global gaze motion and the motion of the pupil, that we have termed the local gaze motion. The global motion captured by the Kinect in 3D includes the translation, in metres, and three angles of 3D head pose, in degrees. The translation is measured by a 3D coordinate with the optical centre of Kinect sensor as the origin, illustrated in Figure 2, while the 3D head pose, measured in degrees, is illustrated in Figure 3, which consists of the pitch, yaw and roll. Whereas the head pose for the face is an important factor in gaze orientation, in this paper we limit the research to the movement of a face orientated forwards. In this paper the global motion consists only of the translational movement and an amalgamation of both translation and rotation under the heading of global motion is currently being researched for future work.

The local motion refers to the pupil motion in planar orbit about two dimensions, X and Y, as compared to the centre of the orbit, shown in Figure 5(a) [12]. In this work it is however assumed that the motions of the left eye and right eye are synchronous.

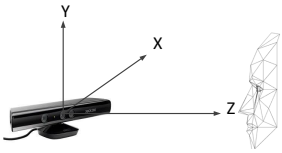


Fig. 2. The 3d space of the Kinect

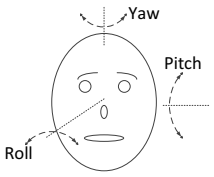


Fig. 3. 3D head pose of pitch, yaw and roll

We summarize the approach via a flowchart that explains the global and local motions shifting the gaze position on the screen (or other objects) in Figure 4.

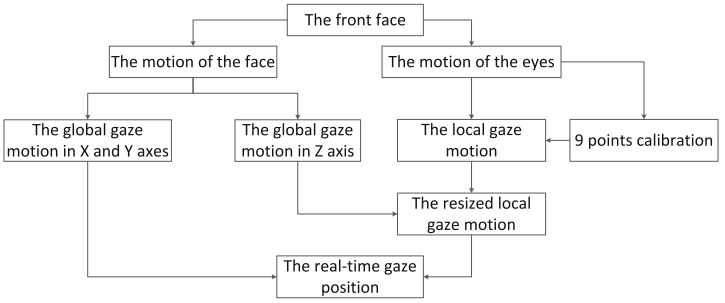
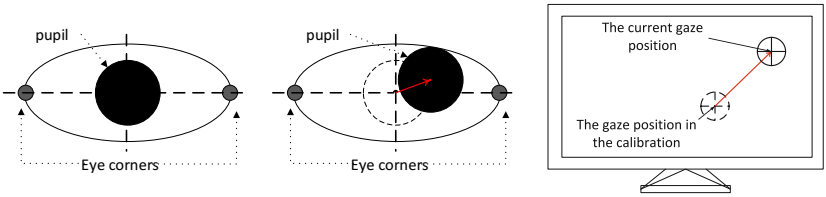


Fig. 4. The flowchart of the proposed model of the gaze estimation

4 Local Gaze Motion

Assuming that the face is fixed and only the local motion of the pupil shifts the gaze position, then the pupil motion can be modelled as the motion in a rectangle, shown in Figure 5(a). While, if the eyes are watching the screen, the screen can be modelled as the other rectangle as in Figure 5(b). So the relation between the gaze position on the screen and the pupil position in the orbit, in two 2D coordinates, can be represented by an affine transformation. In order to get the corresponding position of the pupil in the coordinate system of the screen we need to calibrate two coordinates of the orbit and the screen to obtain its affine transformation matrix. Thus, we require at least 3 matched point pairs to

compute the affine transformation matrix from the 2D coordinates. Therefore, we propose a calibration procedure to get the matched points between the screen coordinate and the orbit coordinate as described in the next section.



(a) The local motion of the pupil in the orbit (b) The gaze shifted by the local motion

Fig. 5. The local gaze estimation model

4.1 Calibration between the Screen and Orbit Coordinate Systems

In the calibration of two coordinate systems, the method of 9 points is the most popular [3], which is also used in the ITU Gaze Tracker. Simply speaking, the 9-point eye calibration process is to sequentially and randomly display the points one by one on the screen and record the positions of the points in the screen coordinate and the corresponding position of the pupil in the orbit coordinate, illustrated in Figure 6. Therefore, we would have 9 matched point pairs between two coordinates after the 9-point calibration. It is important to note that during the 9-point eye calibration the face position should be fixed to avoid any corrupting influence from the global gaze motion.

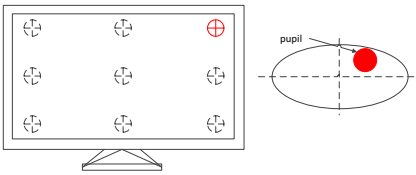


Fig. 6. The 9-point calibration

4.2 Affine Transformation between the Orbit and Screen Coordinates

To compute the affine transformation matrix, 3 matched points pairs are adequate, while from the 9-point calibration we can obtain 9 matched point pairs. Some of the 9 point pairs may not be accurate and would lead to an inaccurate affine transformation. It is necessary to select or obtain 3 accurate point pairs, which are vertices of a triangle, from the 9 points pairs. We propose to exploit two ways to select or construct the 3 points pairs from 9 point pairs in order to ascertain the affine transformation.

1. These 3 point pairs for the affine transformation matrix should be the most common ones, so we can consider that these 3 point pairs are inliers. To select inliner point pairs, RANdom SAmple Consensus (RANSAC) [16] is a popular method. Since RANSAC is a well-known classical algorithm, we will not describe it in detail here and ask the interested reader to consult the reference provided.
2. Since some of the gaze points are considered to be inaccurate, we propose to use the average value to compensate for the inaccurate points. When we display the points on the screen, the positions of the points are exact, but the corresponding points in the orbit coordinates are considered as inaccurate. So we acquire 3 virtual pupil points in the orbit coordinate by averaging 9 pupil points in the orbit coordinate. This procedure is illustrated in Figure 7 and called VIRSELECT by us, and the position of the virtual point $V(m,n)$ is formulated from 9 real pupil points $R(m,n)$:

$$\begin{cases} V(m,n)_x = \frac{\sum_n R(m,n)_x}{3}; \\ V(m,n)_y = \frac{\sum_m R(m,n)_y}{3} \end{cases} \tag{1}$$

where $R(m,n)$ are 9 positions of the real pupil points. m and n are separately 3 rows in y axis and 3 columns in x axis, such that $m = 1,2,3$ and $n = 1,2,3$. $V(m,n)$ are the constructed virtual points.

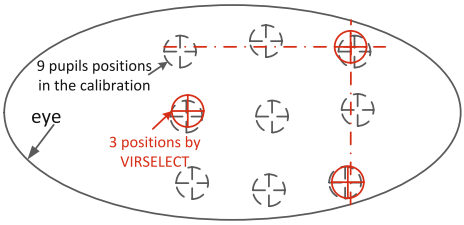


Fig. 7. The principle of VIRSELECT

We compare these two approaches in the experiment described in Section 6.

4.3 Local Gaze Motion

The 2×3 homogeneous matrix M of the affine transformation can be computed by 3 point pairs in the calibration from the previous section, Section 4.2. We then subsequently get the real-time gaze position on the screen $L(x,y)$, caused by the local gaze motion as compared to the coordinate centre in the calibration, obtained from the corresponding pupil position in the orbit $P(x,y)$ by the affine homogeneous matrix.

$$L(x,y) = M \cdot P(x,y,1); \tag{2}$$

5 Global Gaze Motion

In the previous section, we have ascertained the gaze position that is associated with the local gaze motion, which is caused by the movement of the pupil in the orbit. In this section we discuss the global gaze motion shifted by the facial movement. Since we are only concerned with the face translation here we consider the motion of up, down, left, right, forwards and backwards as compared to the position of the Kinect. When a user is looking at the screen, the face is mostly and naturally forwards facing watching the screen in the wide-angle capturing area of the Kinect and the HD Webcam. Therefore, we assume that the face is almost parallel with the screen, the Kinect, and HD Webcam, which means that the pitch, yaw, and roll of the front face in Figure 3 is not obvious.

5.1 The Global Motion in X and Y Axes

When the face together with the eyes moves in X and Y axes as the directions in Figure 2 (up, down, left, and right), the gaze on the screen moves with the same value in the X and Y axes, due to the face being parallel to the screen. The position of the gaze after considering the global motion in X and Y axes following the local gaze motion can be formulated as follows:

$$G_{xy}(x, y) = L(x, y) + K(x, y) \tag{3}$$

where G_{xy} is the current gaze position after considering both the global gaze motion in X and Y axes, $K(x, y)$, and the local gaze motion $L(x, y)$.

5.2 The Global Motion in Z Axis

We have only considered the global gaze motion in the X and Y axes. In this section we discuss the global gaze motion in the Z axis, caused by the face/head moving towards or away from the computer screen. In the frames captured by the HD Webcam and the Kinect, the sizes of the eyes, the pupils and the orbits on the face are resized by the distance between the face and the Kinect or HD Webcam. When the face is nearer to the capturing devices and the screen, the size of the orbits in the frame is larger, and vice versa.

Since the sizes of the pupils and the orbits change together with the distance of the face in the Z axis, the local gaze motion would be different when the face is at a distance that is different then it was at the time of calibration. However, if the relative pupil motion in the orbit is the same as the calibration shown in Figure 8,

$$\frac{p_c}{S_c} = \frac{p_r}{S_r} \tag{4}$$

where p_c and p_r are the pupil motions in a calibration and a real-time test, and S_c and S_r are the sizes of the orbit in the calibration and real-time test, we argue that the local gaze motion in the test is the same as the calibration. Therefore, we can resize the real-time pupil motion in the orbit to its corresponding pupil

motion in the calibration. Thus, we can use the resized pupil motion to get the local gaze motion by the affine transformation matrix obtained at calibration time. Thus Eq. 2 becomes:

$$L'(x,y) = M \cdot P(\frac{x * S_r}{S_c}, \frac{y * S_r}{S_c}, 1) \tag{5}$$

And consequently Eq. 3 can be reformulated as

$$G_{xyz}(x,y) = L'(x,y) + K(x,y) \tag{6}$$

where $G_{xyz}(x,y)$ is the gaze position, according to our proposed model, on the screen considering both the local gaze motion and global gaze motion in the X, Y and Z axes.

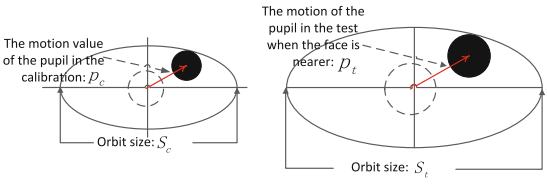


Fig. 8. The pupil motions in the orbit

6 Experimental Results

In this section, we validate the proposed approach of gaze estimation. The experimental arrangement is as shown in Figure 1. The test subjects used for the calibration and the validation sit facing forward to the screen and two capturing devices. During calibration the test subjects are required to maintain a static face position in order to avoid global gaze motion, while during the validation process they are required to randomly move their eyes and their facial position.

Inspired by [15], we propose two simple but effective approaches to evaluate the reliability and the precision of the proposed model of estimating the gaze. Before that, we first compare two methods of RANSAC and VIRSELECT to select 3 points pairs from 9 points for the affine transformation in the local gaze motion.

6.1 RANSAC and VIRSELECT

In Section 4.2 we have proposed two methods, RANSAC and VIRSELECT, to select or construct 3 point pairs from 9 point pairs between the orbit and the screen coordinate systems during calibration. Here we propose a method to validate and compare these two approaches.

The reliability validation is the following: After the calibration, 9 points will be uniformly and sequentially displayed on the screen, each point for 3 seconds, as in Figure 6. The test subject is required to look at the displayed point at that time. If the estimated gaze on the screen is within a circle of diameter 10cm with the displayed point as the centre for more than half of the display time, we consider the gaze reliable to focus on this point, as we need to consider the transition and focusing time for the gaze to move on the screen.

For VIRSELECT, we found that all the 9 points are well focused by the test subjects. However, only the lower 6 points can be focused using RANSAC and the upper 3 points are totally lost. To explain this we argue that the upper area of the screen is more uncomfortable to be focused on than then the lower areas, so the quality of the gaze at the upper area in the calibration is worse than the lower area. Consequently, RANSAC always selects 3 points from the lower area to get the affine transformation. The mean values of the gaze in VIRSELECT compensate the fade zones for the gaze if not moving the face, which are near the corners and the edges of the screen. For these reasons we only use VIRSELECT for the affine transformation.

6.2 Reliability and Precision Validations of Gaze Estimation

The reliability evaluation of the gaze estimation model is the same as described in Section 6.1. However, we present more test data of the gaze falling into differently sized circles with 5 displayed points as the centers on the screen. These 5 points are $(\frac{1}{4}T_w, \frac{1}{4}T_h)$, $(\frac{1}{4}T_w, \frac{3}{4}T_h)$, $(\frac{3}{4}T_w, \frac{1}{4}T_h)$, $(\frac{3}{4}T_w, \frac{3}{4}T_h)$, and $(\frac{1}{2}T_w, \frac{1}{2}T_h)$ where T_w is the width of the screen, and T_h the height of the screen. The reason of using these 5 points is to be far from the fade zones on the screen and to reduce time taken during the experiment – the longer the calibration takes the higher the probability of calibration error due to human error. The reliability validation is shown in Table 1. We measure the reliability by the percentage of the time the gaze falls into the sized circle around the displayed point in the total display duration.

Table 1. The data of reliability validation

Diameter		5cm	10cm	20cm
G_{xy}	center point	6%	14%	68%
	the other 4 points	20%	55%	87.5%
G_{xyz}	center point	19.8%	78.1%	100%
	the other 4 points	33.3%	66.7%	83.3%

In Table 1 we separately show the reliability data of the center point and the other 4 points, where G_{xy} does not consider the distance between the face and the screen and where G_{xyz} does consider this. The reliability values of G_{xyz} are mostly higher than G_{xy}, but the values for G_{xy} become better when the circle diameter is larger because the gaze is still nearby without considering the face

motion in the Z axis. We can see that the reliability data on the center point is mostly higher than the other 4 points, because the center point is further from the fade zones around the edges and the corners of the screen. When the diameter is 5cm, the reliability of the center point is less than the other points, which is caused by averaging and compensating the low reliability values by lots of data from 4 points.

For the precision we consequently show 6 points in a triangle, shown in Figure 9. The reason of displaying fewer points in the lower area is to avoid the pitch action of the face. Each point is shown for a short time, approximately 0.5 seconds. If the gaze can hit a point within the short time to within a tolerance of 5cm or 10cm diameter, the precision value is increased by 1/6. In Table 2 we show the precision of G_{xyz} and G_{xy}. It can be seen that the precision of the gaze estimation model is very high.

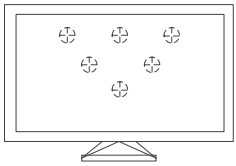


Fig. 9. The precision validation

Table 2. The data of precision validation

Diameter	5cm	10cm
G_{xy}	66.7%	100%
G_{xyz}	83.3%	100%

Comparing our results with those from the literature [15] our proposed method performs exceptionally well for the reliability and precision. Compared to [23], we can see that we achieve accurate gaze tracking too considering the distance to the capturing devices. Since the hardware and facing problems are so distinguished, it is impossible to make the quantitative comparison. It should be noted that in our method the motion of the face in the X and Y axes should be less than 7cm because due to the capturing ability of the Kinect and HD webcam. Additionally the face motion in the Z axis should be limited in a range of 20cm as has been detailed in [12].

7 Conclusions and Future Work

In this paper we have proposed a real-time gaze estimation model based on simple capturing devices for a desktop computer user. We have separated the gaze motion on the screen into local motion and global motion. The local motion

refers to the gaze motion driven by pupil movement and the global motion to motion driven by head movement. By analysing the face movement in the Z axis, moving towards and away from the screen, our proposed method resizes the orbit together with the motion of the pupil, and correspondingly rescales the local gaze motion. Our proposed gaze estimation model uses only simple capturing devices, a HD webcam and a Kinect, but can achieve real-time and accurate gaze estimation, as demonstrated by the presented experimental results.

In our future research we are working to expand our method to incorporate the 3D head pose rotations of a user, i.e. the yaw, pitch and roll. By incorporating the user head rotations into our method we hope to make our gaze estimation model more robust to account for naturalistic human head motion and also to allow for larger computer displays where head rotation is needed to view around the screen. The current proposed system is only for one person, but Kinect and ITU gaze tracker is probable to process multiple faces. Therefore, we are considering the probability of the proposed system for multiple heads.

Acknowledgement. The research that lead to this paper was supported in part by the European Commission under the Contract FP7-ICT-287723 REVERIE.

References

1. Duchowski, A.T.: A breadth-first survey of eye-tracking applications. Behavior Research Methods, Instruments & Computers 34(4), 455–470 (2002)
2. Hansen, D.W., Ji, Q.: In the eye of the beholder: A survey of models for eyes and gaze. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(3), 478–500 (2010)
3. Duchowski, A.T.: Eye tracking methodology: Theory and practice, vol. 373. Springer (2007)
4. Morimoto, C.H., et al.: Eye gaze tracking techniques for interactive applications. Computer Vision and Image Understanding 98(1), 4–24 (2005)
5. Bohme, M., Meyer, A., Martinetz, T., et al.: Remote eye tracking: State of the art and directions for future development. In: Proc. of the 2006 Conference on Communication by Gaze Interaction (COGAIN), pp. 12–17 (2006)
6. Wang, J.G., et al.: Eye gaze estimation from a single image of one eye. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, pp. 136–143 (2003)
7. Kim, K.N., Ramakrishna, R.S.: Vision-based eye-gaze tracking for human computer interface. In: IEEE SMC 1999 Conference Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, pp. 324–329. IEEE, MLA (1999)
8. Tan, K.H., et al.: Appearance-based eye gaze estimation. In: Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision, pp. 191–195. IEEE (2002)
9. Reale, M., et al.: Using eye gaze, head pose, and facial expression for personalized non-player character interaction. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 13–18 (2011)

10. Langton, S.R.H., Honeyman, H., Tessler, E.: The influence of head contour and nose angle on the perception of eye-gaze direction. Perception & Psychophysics 66(5), 752–771 (2004)
11. Funes Mora, K.A., Odobez, J.-M.: Gaze estimation from multimodal Kinect data. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 25–30 (2012)
12. Li, Y., Wei, H., Monaghan, D.S., OConnor, N.E.: A Hybrid Head and Eye Tracking System for Realistic Eye Movements in Virtual Avatars. In: The International Conference on Multimedia Modeling (2014)
13. Jafari, R., Ziou, D.: Gaze estimation using Kinect/PTZ camera. In: IEEE International Symposium on Robotic and Sensors Environments (ROSE), pp. 13–18 (2012)
14. Andrist, S., Pejsa, T., Mutlu, B., Gleicher, M.: A head-eye coordination model for animating gaze shifts of virtual characters. In: Proceedings of the 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction (2012)
15. Ciger, J., et al.: Evaluation of gaze tracking technology for social interaction in virtual environments. In: Proc. of the 2nd Workshop on Modeling and Motion Capture Techniques for Virtual Environments (CAPTECH 2004) (2004)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
17. Villanueva, A., Cabeza, R.: Models for gaze tracking systems. Journal on Image and Video Processing 2007(3), 4 (2007)
18. Li, D., et al.: openEyes: A low-cost headmounted eye-tracking solution. In: Proceedings of the ACM Eye Tracking Research and Applications Symposium (2006)
19. Nussbaum, G., Veigl, C., Acedo, J., et al.: AsTeRICS-Towards a Rapid Integration Construction Set for Assistive Technologies. In: AAATE Conference (2011)
20. Zielinski, P.: Opengazer: open-source gaze tracker for ordinary webcams (software), Samsung and The Gatsby Charitable Foundation, `http://www.inference.phy.cam.ac.uk/opengazer/`
21. Savas, Z.: TrackEye: Real time tracking of human eyes using a webcam, `http://www.codeproject.com/KB/cpp/TrackEye.aspx`
22. San Agustin, J., Skovsgaard, H., Hansen, J.P., et al.: Low-cost gaze interaction: ready to deliver the promises. In: CHI 2009 Extended Abstracts on Human Factors in Computing Systems, pp. 4453–4458. ACM (2009)
23. San Agustin, J., et al.: Evaluation of a low-cost open-source gaze tracker. In: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications. ACM (2010)