A Low-cost Head and Eye Tracking System for Realistic Eye Movements in Virtual Avatars

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

INSIGHT Centre for Data Analytics, Dublin City University, Ireland

Abstract. A virtual avatar or autonomous agent is a digital representation of a human being that can be controlled by either a human or an artificially intelligent computer system. Increasingly avatars are becoming realistic virtual human characters that exhibit human behavioral traits, body language and eye and head movements. As the interpretation of eye and head movements represents an important part of nonverbal human communication it is extremely important to accurately reproduce these movements in virtual avatars to avoid falling into the well-known "uncanny valley". In this paper we present a cheap hybrid real-time head and eye tracking system based on existing open source software and commonly available hardware. Our evaluation indicates that the system of head and eye tracking is stable and accurate and can allow a human user to robustly puppet a virtual avatar, potentially allowing us to train an A.I. system to learn realistic human head and eye movements.

Keywords: Avatar, Eye tracking, Head movement, Uncanny valley

1 Introduction

In this paper we present a hybrid head and eye tracking technique for use in both understanding and reconstructing realistic human head and eye movement in virtual avatars. An avatar, put simply, is any graphical representation of a person or user. This representation can take many forms: from simple icons, personalized cartoon characters, pictorial mock ups to full 3D humanoid representations. Often in computer gaming, virtual avatars can be fictional or fantasy based characters where a user will personalize the avatar based on how that user wishes to be represented. However, recently there has been a shift towards realistic personal avatars that accurately represent the user. This shift has been brought about, in part, by readily available and cheap data capture platforms such as the Microsoft Kinect depth sensor.

In the fields of animation and robotics there is a commonly used term known as the "uncanny valley" [1], which describes a sharp dip in familiarity or comfort levels that humans have towards virtual humans the closer they come to mimicking or looking like real human beings. This refers to the phenomenon whereby we typically feel very comfortable with a virtual avatar that we know for certain is a cartoon or animation, however we become very uncomfortable with a virtual avatar that is almost, but not quite, human. Thus it is important to ensure a virtual avatar exhibits realistic and believable human features and behaviors. Among the most important of these features and behaviors are realistic facial, head and eye movements, as these contribute significantly to nonverbal communications. Indeed, tracking only one of these, e.g. tracking the head but ignoring eye movement within the eye socket, will lead to visually disturbing virtual representations. To address this, in this paper we propose a simple system for human eye tracking using a common HD webcam and a Microsoft Kinect. The cheap hardware and open source software toolkits used are currently widely available and we fuse them in a simple and efficient way to achieve real-time performance. To the best of our knowledge, the application of open-source and non-commercial systems for resolving the problem of eye tracking and thereby helping to avoid the "uncanny valley" has not been widely investigated in the state of the art.

The proposed system is low-cost, real-time and accurate. The cost of the hardware is around 200euro of a Kinect for Windows and a Logitech Webcam, which requires much less budget than the normal commercial system of eye tracking. Furthermore, this system could achieve real-time performance and satisfy the real-time requirement. And in the given environment, the detected eye center is only 1-2 pixel away from the real eye center in the captured video.

2 An Overview of the Proposed Approach

We differentiate three aspects of our approach: the macro module, the micro module and the fusion module. The first deals with robust head tracking by utilizing the Microsoft Kinect depth sensor. In this way, the head position, orientation and gaze direction can all be approximated with a high level of accuracy. The second module then leverages this to perform eye tracking utilizing a modified version of the open source ITU Gaze Tracker. For this we use a cheap high definition (HD) USB web camera (Logitech 1080p, whose price is around 70euro). The eye or gaze direction can be obtained by approximating the position of the pupil and the two corners of the eye. It should be noted that the RGB camera integrated into the Kinect hardware was found, via experimental observation, to be of too low a quality and resolution to be used robustly for the purpose of eye tracking within our scenario. And this is also the reason to exploit an additional HD Webcam. The outputs of these two modules are integrated and the pupil position and head movement and orientation are subsequently fused to puppet a human controlled avatar. The proposed approach is illustrated in Fig. 1. We describe each module in detail in the following sections.

3 Eye and Gaze Tracking

The believability and realism of virtual autonomous agents/avatars will, in part, depend on human-like gaze (eye) movement and attention within a virtual environment. Furthermore, when a human-avatar interaction is taking place the gaze of an avatar should be able to mimic the human operator. Visual attention

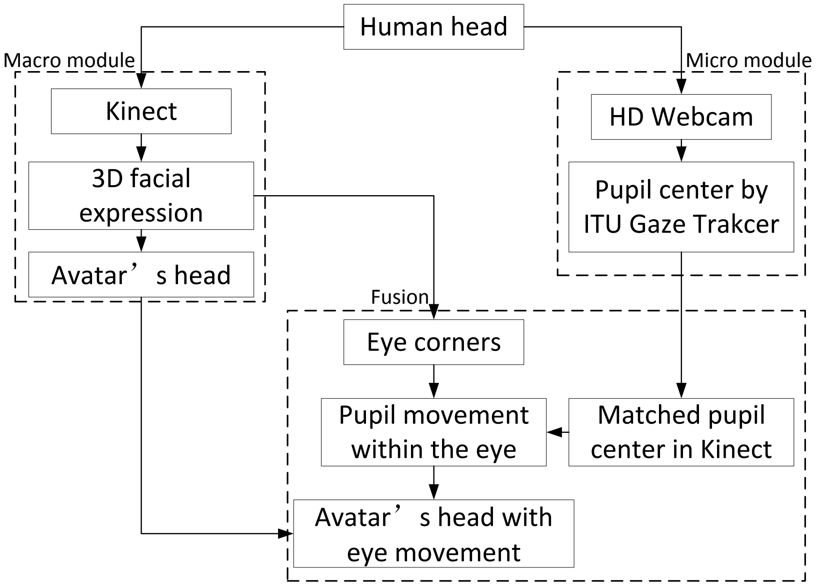

Human head
Macro module
Kinect
3D facial expression
Avatar's head
Micro module
HD Webcam
Pupil center by ITU Gaze Trakcer
Fusion
Eye corners
Pupil movement within the eye
Matched pupil center in Kinect
Avatar's head with eye movement

Fig. 1. Overview of the proposed approach

has been widely researched for over 100 years. The study of this originated from the field of psychology has now gained popularity within the field of computer science and in particular, Human-Computer Interaction (HCI). The fundamental goal of eye and gaze tracking is to estimate the direction of gaze of a person [3], and its output should be an estimation of the projected point of view with regards to the person [2]. An ideal eye gaze tracker should be accurate, reliable, robust, non-intrusive, operate as close to real-time as possible, be robust to head motion vibrations, and not require complex calibration procedures [3].

Depending on the application, Eye and Gaze Trackers (EGTs) are separated into two groups, diagnostic applications and interactive applications [4]. In diagnostic applications EGTs are used to quantify a user's attention, while in interactive applications EGTs are used to interact with the user for use in a device or computer [3]. According to the distance between the EGT capture device and the human subject's eyes, EGT systems are categorized into Remote Eye and Gaze Tracking systems (REGT) and Head Mounted Eye and Gaze tracking systems (HEGT). For the purposes of our work we are concerned primary with REGT systems that are non-invasive and that do not require expensive and fragile head mounted hardware. It should, however, be noted that the current state-of-the-art REGT systems demonstrate the significant challenges to be addressed, specifically with respect to head motion and they typically exhibit lower accuracy than HEGT. By using our novel hybrid system that combines robust head tracking and eye tracking we overcome some of these existing problems with REGT approaches, for example the fixed head position and pose, and the expensive hardware.

The ITU Gaze Tracker [5][6] is a camera based eye and gaze tracking system that is low-cost, robust, accurate, has flexible components, and is open source. It tracks the pupil center and corneal reflections (i.e., glints) when utilized with infrared light. The main principle behind its pupil center extraction method is to identify and extract the points in the contour between the pupil and the iris, which are subsequently fitted to an ellipse using the RANSAC algorithm in order to eliminate outliers. There are several native tracking modes within the ITU Gaze Tracker: Headmounted, Remote Binocular and Remote Monocular, which are designed for use with both HEGT and REGT systems, however the stability of these tracking modes varies. HEGT tracking has a distinct advantage over the REGT modes owing to the fact that when a camera is placed nearer to the eye the spatial sampling rate or quality of the captured image is higher than when the camera is placed a further distance from the eye. We customized different aspects of these tracking modes to suit specifically to our needs and employed a low-cost HD webcam, the Logitech C920, to build a REGT setup. Using a HD webcam we can accurately focus on the human face and capture a good quality image of the face and the area around eyes.

Fig. 2 shows an example of the combination of facial detection and eye tracking from the eye tracker when the subject is 40-50cm away from the webcam. It can be seen that we can obtain good image quality and pixel resolution, and that the eyes are well focused by the HD webcam. Fig. 3 shows a close up of the subject's eyes.

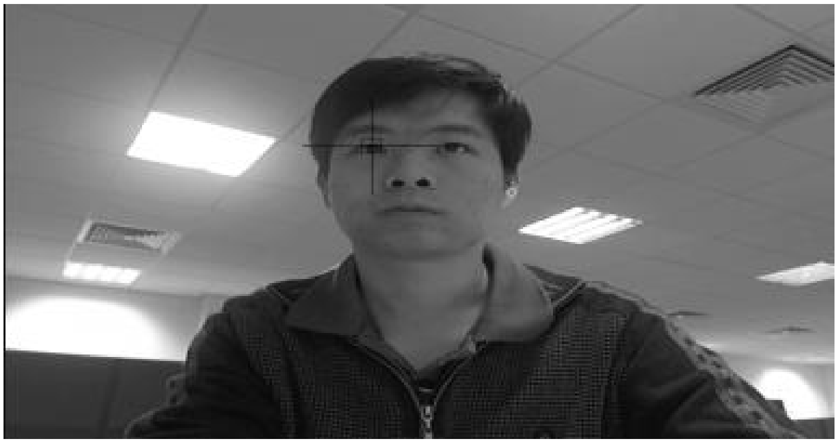


Fig. 2. Eye location identification utilizing a modified version of the ITU Gaze Tracker for REGT

The ITU Gaze Tracker mode used in our approach does not consider if the detected pupil is from the left eye or right eye. To facilitate this we created and

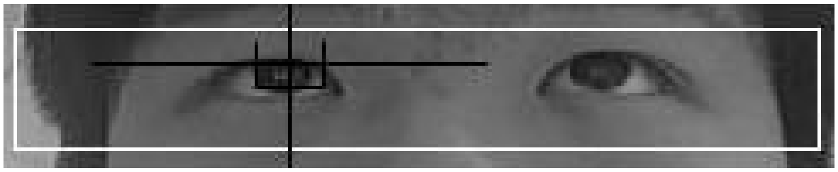


Fig. 3. Close up segmentation of the tracked eye pair

integrated a function that detects the presence of an eye pair by means of a Haar Cascade [7], shown in Fig. 3 by the white rectangle. Then the position of the detected pupil in the eye pair, nearer to left border or right border, is used to calculate whether it is a left or right eye.

Alongside the modified and new functions added to the eye tracker, additional data outputs are leveraged in our hybrid approach. This computed data includes, left/right pupil tracking, pupil pair identification, pupil position, region of interest around pupil as well as the native ITU Gaze Tracker outputs of time stamp, and gaze position. This additional data is fed into the hybrid tracking system and used to robustly combine the eye tracking and head tracking systems outputs.

4 Head Position and Orientation Estimation

Head pose estimation [8] and facial expression tracking [9][10] have recently received much attention in the literature, in particular in fields of research pertaining to human-computer interaction, facial animation, virtual avatar creation etc. Facial expression analysis provides the important features and data on the human face which enable the further study of face animation and realistic virtual avatar creation.

Within the context of this work we define head pose estimation to be the estimation of the orientation of a person's head in relation to the position of the camera being used to capture that person. Head pose estimation is also used as an initial input into gaze tracking algorithms because it can offer a good estimation of the gaze direction.

To the best of our knowledge, most research on head pose and facial expressions are based on two dimensional image processing, which contains few distinct facial features and can be easily influenced by the capturing environment. However, the emergence of the low cost Kinect depth sensor platform provides us with a new and cheap platform to easily ascertain head pose and facial expression information [11] [12]. The Kinect consists of an RGB camera, an infrared (IR) emitter and an IR camera, and by using a reconstruction process known as the deformation of structured light, it can capture the depth of a scene with an effective range of 0.45 - 7.0 meters. The Microsoft Kinect toolkit provides a 3D face model and its corresponding data, shown here in Fig. 4(a) and 4(b).

In this work we use the Kinect for Windows system in the near mode, which enables us to capture the face with the minimum distance of 40cm between

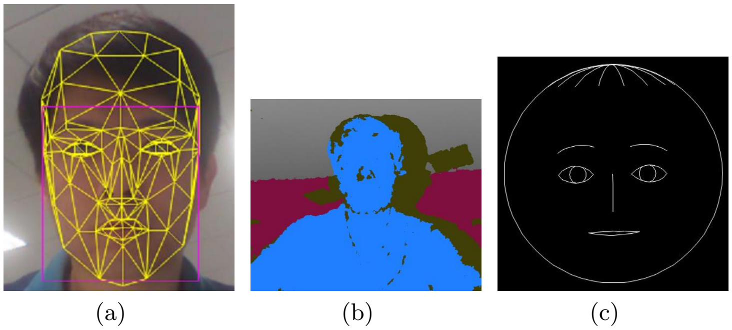


Fig. 4. (a) 3D facial tracking model from Kinect; (b) Depth map image; (c) A sample avatar's head from the face tracking toolkit

the person and the Kinect, while Xbox Kinect needs a much further minimum distance of approximately 1.2 meters. The output data from the Kinect face tracking can be used in facial animation, as the position of eye corners are very useful in eye and pupil tracking, as will be explained in Section 5. The Kinect face tracking visualization, shown in Fig. 4(c), can demonstrate the animation of an avatar's head corresponding to the motion of the person in real time and this forms a good basis for a hybrid tracking system. Furthermore, in Fig. 4(c) we can see that the pupil size in the orbit is too large to move in the orbit, so we shrink the pupil size in the avatar's head. We also enable this module to receive the additional data through UDP from the eye tracking module.

5 Hybrid Head and Eye Tracking System

In Section 3 we presented eye and pupil tracking and in Section 4 we presented head and facial tracking. Both of these systems have been uniquely modified, improved and adapted in order to fuse the data and animate the eyes in an avatar's head using a hybrid tracking system. In this section, we describe our novel approach to achieve real-time head and eye tracking.

The modified avatar head from Section 4 responds well to head movements and facial expression but the pupils cannot move. If we want to animate the pupils, we need to know the movement of the pupils from the subject at any given point in time. From Section 3, we only know the position of the pupil and the movement of the pupil in the context of the head is not known. To address this we firstly analyze the movement of the pupil. As shown in Fig. 5(a), we can see that the pupil is normally in the center of the eye socket. When the pupil moves, as shown in Fig. 5(b), there is a distance between the pupil and the center of the socket, which can be defined by eye corners (canthus).

From Section 3, we know the real-time position of the pupil, so the positions of eye corners are required. Some successful algorithms for the detection of eye corners [13] [14] already exist, however, many of them are implemented offline

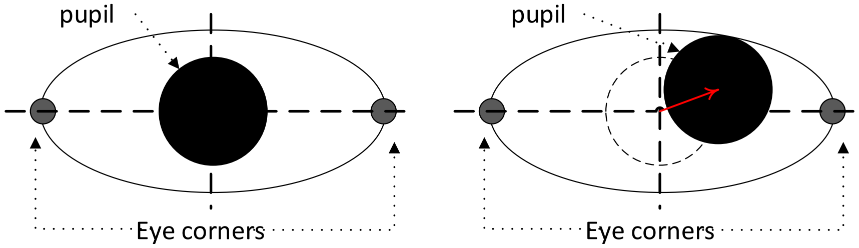


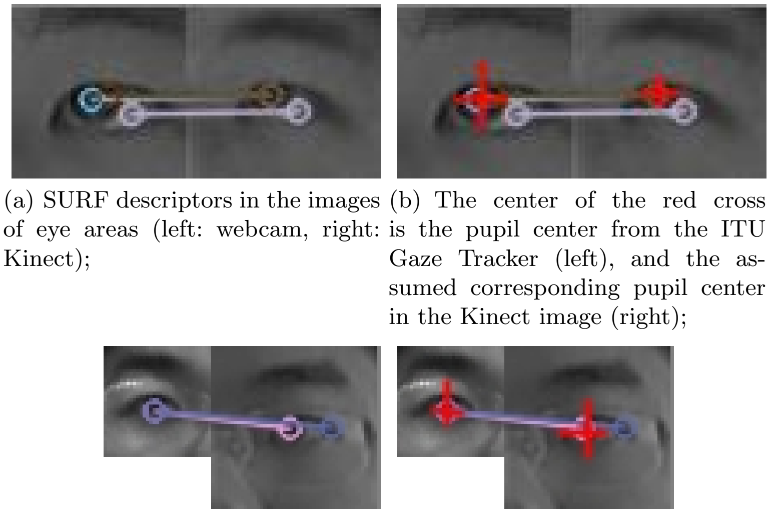
Fig. 5. The model of pupil motion: (a) The pupil is in the eye center; (b) The pupil moves away from the eye center

and the others cannot promise the robustness that is required in this system. In our case, the robust and real-time coordinates of eye corners are required. We have found that the Kinect can provide this kind of data and satisfies our criteria for robust and real-time processing.

As previously mentioned, it was found that the RGB camera from Kinect is of too low quality to be robustly used for eye tracking, so we use the eye information from a separate HD web camera, as described in Section 3. To handle rectification between the two camera systems it is typical to find the corresponding pixels by homography and camera calibration. However, we have found that the accurate homography computations would not allow for the system to operate in real-time. On the other hand, if isolated segmentation of an eye pair from each camera system is used, real time performance is achieved but insufficient matching points (via SURF descriptors [16].) for the computation of a homography [15] are obtained. It is a challenge for homography computation to find a good compromise between computation time and sufficient SURF descriptors. Camera calibration before eye tracking is of course possible but this makes the hardware setup more complex and less accessible in general.

The SURF algorithm can detect distinctive locations in the image, such as corners, blobs, and so on. Since the pupil is a very distinctive blob in the eye images, the pupil center is one of the detected key points, as shown in Fig. 6(a). For the image from Kinect we need to sharpen the image as a preprocessing step to detect SURF descriptors, because the image from Kinect is not focused and is often blurred. The image size of eye region from webcam is $41 - 51$ pixels for both width and height and the width and height of eye region image from Kinect camera is set as twice the distance of two eye corners.

From Fig. 6(a) we can see that pupil centers are detected as a good match of SURF points. The eye tracker from Section 3 provides us the pupil center in the webcam image, then the SURF point nearest to it is the detected blob center (pupil center) in the webcam image and the matched point in Kinect image is the corresponding pupil, illustrated in Fig. 6(b). The distance between the detected blob center and the pupil center by eye tracking is 0-2 pixels in Fig. 6(b) left, then in Fig. 6(b) right the distance between blob center and the assumed pupil center should be less than 1 pixel, because of a smaller eye captured by the wide-angle RGB lens of Kinect. Therefore, we can get an accurate pupil center in the Kinect image.



(a) SURF descriptors in the images of eye areas (left: webcam, right: Kinect);

(b) The center of the red cross is the pupil center from the ITU Gaze Tracker (left), and the assumed corresponding pupil center in the Kinect image (right);

(c) Another example of the face with the glasses;

(d) The matched pupil center of another example in (c)

Fig. 6. SURF matched points

Consequently, we can determine the pupil movement within the eye as illustrated in Fig. 5 and then we can animate the pupil on the avatar's head in real time, 4 snap shots of which are shown in Fig. 7 to show the effects in different head poses and eye positions. A demonstration video can also be viewed here: http://goo.gl/b0bhD. With the animated eye on the face, it is clear to see that the virtual avatar is more like a human face than the face without the eye motion in the face tracking demo of Kinect SDK. In the sense of eye motion, we partially improve the solution for "uncanny valley".

6 Evaluation

Fig. 7 and its corresponding video provide a subjective evaluation of the proposed approach. Objective evaluation, on the other hand, is not straightforward. Benchmarking against existing head mounted or remote systems is difficult as typically the operation of such systems will interfere with our set-up meaning that it is difficult to perform tracking simultaneously using two different systems. Thus, for objective evaluation we compare the tracked pupil centers in the

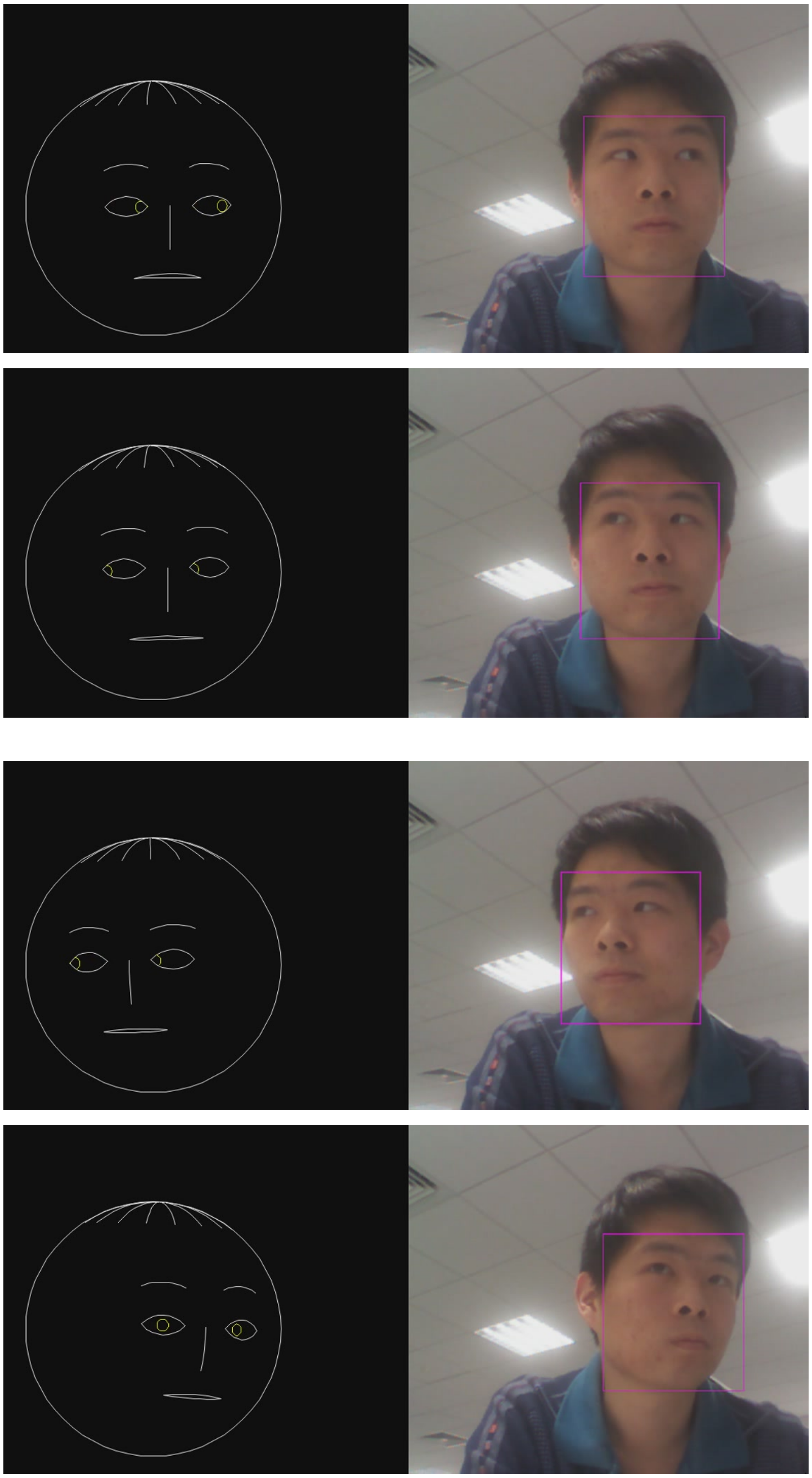
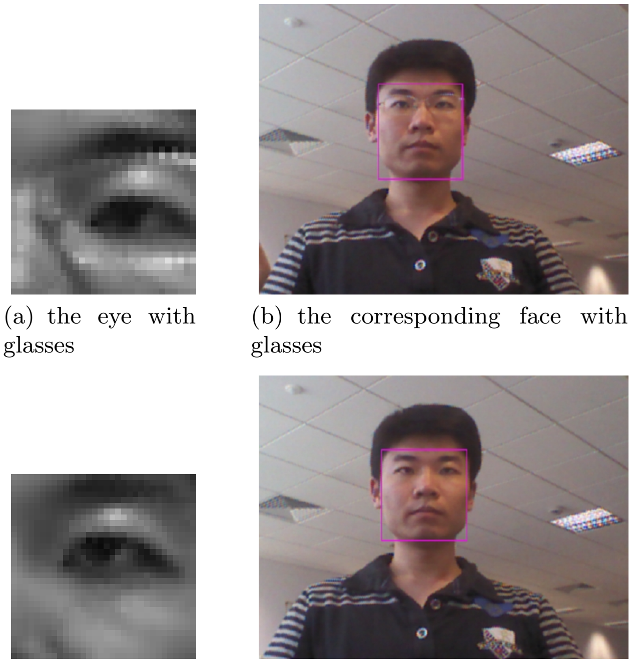


Fig. 7. Examples of avatar's head with animated eyes for various head and eye positions

Kinect video frames by our system with the pupil centers manually annotated by human beings.

We asked 4 people to select the pupil centers from 700 Kinect frames in total, captured from 2 different people, with and without glasses. Glasses typically pose challenges for most existing tracking systems. We require our system to be robust to the presence/absence of glasses, or even to glasses being removed/introduced. The example frames of eye regions with and without glasses are shown in Fig. 8. As our goal was to obtain a sample of manual annotations that was as representative as possible, each annotator could choose some subset of frames, e.g. ignoring very similar frames, with the constraint that at least 50 pupil centers should be selected by each annotator. We then calculate the Euclidean distance between the manually selected centers and the centers as recorded by our system.



(a) the eye with glasses

(b) the corresponding face with glasses

(c) the eye without glasses

(d) the corresponding face without glasses

Fig. 8. The eye regions with and without glasses

The Euclidean distances, with and without the glasses, are shown in Table 1. The mean values of the distances between the pupil centers by the proposed system and the ground truth are 2.46 pixels and 2.87 pixels. Considering that it is hard for people to exactly pick the pupil centers to the accuracy of a single pixel this indicates good performance. Furthermore, the distances of the pupil centers at around 2.5 pixels, compared to the size of the eye region at around 41-51 pixels, are accurate. Finally, the proposed system performs the same no matter if the person wears the glasses or not, so the proposed system is stable in this regard.

Table 1. The Euclidean distance between the human annotated pupil centers and the centers tracked by our system in Kinect frames

(pixel)	Person 1	Person 2	Person 3	Person 4	Mean value
with glasses	2.8438	2.7699	2.1916	2.0407	2.46
without glasses	2.8937	3.0857	2.8496	2.6514	2.87

In the experiments, we found that when the face is at the distance of 40-60cm away from the webcam, ITU gaze tracker can provide the real-time eye image to the proposed system. When the face is further, the higher resolution of the webcam frame for ITU gaze tracker is necessary, but PC in the experiment cannot perform the real-time processing for the computation of ITU gaze tracker. Therefore, a further distance is possible for the real-time and robust performance of the proposed system, but a higher standard PC is necessary.

7 Conclusion

In this paper we have proposed a hybrid system for capturing eye and head motions by fusing together modified Kinect and HD webcam systems. The system runs in real time and is demonstrated by animating the head and eye in a customized Kinect avatar. The proposed system is low cost and easy to setup. A demonstration video shows our proposed system working in real time and a subjective evaluation shows that the system is robust and accurate for situations when both eye and head are in different positions or poses. Furthermore, we have measured the accuracy of the proposed hybrid system against a human annotated ground truth. Thus, we have gone some way towards resolving the specific problems associated with pupetting eye movements in virtual avatars in a realistic manner, which is a novel and important application of our proposed system. The head and eye movements could in the future be recorded and fed into a machine learning classifier to train an artificial intelligent computer system to control a virtual avatar, using features from both head and eye movements. This is a target for our future work.

8 Acknowledgements

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

References

1. Mori, M, "The uncanny valley. Energy", 1970, 7(4): 33-35.
2. Duchowski, Andrew T, "Eye tracking methodology: Theory and practice", Vol. 373. Springer, 2007.
3. Morimoto, C.H. and Mimica, M.R.M., Eye gaze tracking techniques for interactive applications, Computer Vision and Image Understanding, Vol. 98, No. 1, Pages 4-24, 2005.
4. Brolly, X.L.C. and Mulligan, J.B., Implicit calibration of a remote gaze tracker, Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop, Volume 8, p. 134, 2004.
5. San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D. W., and Hansen, J. P, Evaluation of a low-cost open-source gaze tracker, In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, 2010.
6. ITU Gaze Tracker: http://www.gazegroup.org/downloads/23-gazetracker
7. Modesto Castrilln-Santana and O. Dniz-Surez, L. Antn-Canals and J. Lorenzo-Navarro, Face and facial feature detection evaluation, in Third International Conference on Computer Vision Theory and Applications, VISAPP08, 2008.
8. Murphy-Chutorian, E. and Trivedi, M.M., Head pose estimation in computer vision: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 31, number 4, pages 607-626, 2009.
9. Shan, C. and Gong, S. and Mcowan, P.W., Facial expression recognition based on Local Binary Patterns: A comprehensive study, Image and Vision Computing, volume 27, number 6, pages 803-816, 2009.
10. Terzopoulos, D. and Waters, K., Physically-based facial modelling, analysis, and animation, The journal of visualization and computer animation, volume 1, number 2, pages 73-80, 1990.
11. Cruz, L. and Lucio, D. and Velho, L., Kinect and rgbd images: Challenges and applications, SIBGRAPI Tutorial, 2012.
12. Zhang, Z., Microsoft Kinect Sensor and Its Effect, Multimedia, IEEE, volume 19, number 2, pages 4-10, 2009.
13. Santos, G. and Proenca, H., A robust eye-corner detection method for real-world data, Biometrics (IJCB), 2011 International Joint Conference on, 2011.
14. Xu, C. and Zheng, Y. and Wang, Z., Semantic feature extraction for accurate eye corner detection, ICPR, 2008.
15. Sukthankar, R. and Stockton, R.G. and Mullin, M.D., Smarter presentations: Exploiting homography in camera-projector systems, ICCV, 2001.
16. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2006.