

REAL-TIME GAZE TRACKING WITH A SINGLE CAMERA

Constantin Carapencea ^{*1}
Irina Mocanu ²

ABSTRACT

This paper presents a non-intrusive method for gaze tracking, using a single monocular low-resolution camera. By tracking the face orientation and the center of the pupil, the system estimates the gaze direction which can be projected onto a screen to obtain the user's point of gaze. As the design is based on open source portable building blocks, the system can be used on a variety of devices. The implemented gaze tracker achieves real-time performance. Although the measured accuracy does not compete with other expensive, more complex solutions, it is still possible to integrate it into other applications which require basic gaze tracking capabilities.

Key-words: GAZE, monocular low-resolution camera, gaze tracking, human-computer interaction

1. Introduction

Eye gaze tracking is the process of automatically tracking a subject's gaze. This process has recently been integrated in many applications in various fields. In human-computer interaction, the information about the user's gaze direction can be used to monitor areas of interest, to control various actions or to learn personalized attention distribution patterns in order to provide an improved user experience. This can lead to more intuitive intelligent interfaces, as the process of estimating gaze direction is natural for humans. Such interfaces can even allow disabled people with limited motion capabilities or severe paralysis, who can still use their eyes, to interact with computers or other digital devices [1].

Psychology can benefit from using gaze tracking information to study both perceptual and cognitive processes. For example, one could use this information to study how humans perform visual search tasks [2]. Combining this knowledge with real-time machine interfaces opens the path towards important applications such as traffic safety devices which monitor a driver's gaze to make sure she is paying attention to the road and provide certain notification signals otherwise in order to prevent accidents caused by driver fatigue [3], [4].

Depending on the application specifics, the gaze can be measured as the point of gaze, or as the gaze vector, whose direction is estimated by the line passing through the fovea (the region near the center of the retina with the highest density of cone cells, responsible for sharp vision) and the center of the pupil [5]. The intersection of this vector with the subject's environment gives the point of gaze which can be tracked in order for a system to be able to predict the user's intentions or to provide a means to easily control subsystems, such as turning devices on or off in an intelligent house by simply fixating

¹ * Corresponding author. University POLITEHNICA of Bucharest, Splaiul Independentei No. 313, Bucharest 060042, Romania. Email: constantin.carapencea@cti.pub.ro

² Associate Professor, PhD, University POLITEHNICA of Bucharest, Splaiul Independentei No. 313, Bucharest 060042, Romania. Email: irina.mocanu@cs.pub.ro

them. The point of gaze need not necessarily be three-dimensional: for example certain applications could track the point a person is gazing at on a screen which can be encoded in only two dimensions.

The rest of the paper is organized as follows. Section 2 presents some existing methods for gaze tracking. Section 3 describes the proposed approach, while details about the experimental results of the proposed method can be found in Section 4. Conclusions and future work are presented in sections 5.

2. Related Works

In order to minimize both the intrusiveness and the cost of such methods, a considerable amount of effort has been made to develop various technologies based on analyzing digital images of the person whose gaze is to be tracked, captured with cameras of various performances. this approach is still very attractive due to its minimal setup requirements and reduced development and installation costs, as such a system can be implemented using only a single commodity web camera such as those already available in many consumer devices such as laptops, tablets and smartphones. Bellow are revised several existing methods used for gaze tracking.

Gaze Estimation Using the Face Normal Vector: Having identified 4 key points (mouth corners and eye corners) in a still image or a video frame containing a face, it becomes possible to determine the orientation of the subject's head (face plane normal vector) using a simplified face model. The skew of these 4 points in the face plane can be used to recover the rotation. For small rotation angles, when the skew is less visible (i.e. the face plane is almost parallel to the image plane) an additional point on the tip of the nose can be used to create a 3D model and improve accuracy, assuming a weak projection camera model [6]. One limitation of this method is the fact that it does not take into account the eyeball rotation independent of the head rotation. On the other hand, in conjunction with an imaging technique which could estimate the eyeball rotation relative to the user's head, it could be possible to increase the accuracy.

Computing Gaze Vector Based on a 3D Head and Eye Model: This method is based on establishing a three-dimensional model of the head and eyes. For example the head is modeled as a sphere on which there are two fixed positions where the eyeballs are located. The two eyeballs can also be approximated as spheres. Based on this model, it is possible to compute the gaze vector if the position and orientation of the head and the orientation of the eyeballs relative to the eye sockets are known. As a result this approach implements a three-step procedure. Firstly, the orientation and position of the head is estimated. Secondly, the orientation of the eyeballs is estimated relatively to the local features such as the eye corners. Finally, these measurements are plugged into the model in order to compute the gaze vector [3]. Many applications require the detected gaze direction to be relative to a certain fixed point (e.g. device screen), therefore an increased precision is required for computing the head orientation. This is a key problem where a gaze tracking system could benefit from stereo imagery. By using two cameras to capture images of the subject, after detecting the facial features and triangulating their 3D position, a reasonably accurate 3D model of the face can be constructed [7]. Such a system could also use a slightly more accurate model of the eyeball as the pupil is actually located on a sphere and not on the same fixed plane as the eye corners [8]. Although the stereoscopic hardware is more complex, an ad-hoc implementation is possible using two commodity USB web cameras, but, in this case, there are additional requirements such as camera stereo

calibration, screen position calibration and making sure the cameras and the screen remain in the same exact position during a certain session [8].

Infra-Red Lighting: By illuminating the users face using IR light (e.g. from a LED) two phenomena are observed which prove beneficial for gaze tracking applications: the bright pupil effect provides a solid contrast between the pupil (very bright) and the iris (dark) which can considerably improve the accuracy of the tracking; a glint is caused by the reflection of the IR light from the cornea, which, if tracked alongside the pupil, can provide the second point of the gaze vector. More accurate 3D models can be constructed either using multiple cameras (similar to the visible light stereo tracking), or by using a single higher performance camera and multiple alternating IR light sources synchronized with the camera frame rate [9]. Despite of the advantage observed in tracking simplicity, this method is considerably more expensive (lights, specialized cameras) and more difficult to set up.

Conic Iris Projection: This approach is based on the elliptical features of the iris and it extracts the normal vector from the projection of the iris in the captured image. The main steps are finding the contour of the iris in the image plane and fitting an ellipse in the general case to the points on this contour in order to get the ellipse's parameters. Furthermore, this ellipse is the result of projecting a circle (the iris) from its gaze plane (the plane whose normal is the gaze vector) to the image plane through the optical center of the camera. By parameterizing both cones, one with the known ellipse parameters, and the other with the gaze vector, the latter parameter can be computed using the images of the both eyes [10]. Provided high resolution images, in which the iris contour can be detected with a reasonably small error, this method can provide impressive results with accuracy as good as 0.8° [11].

Face Tracking in Image Based Gaze Estimation: The requirement that the system should work with low resolution images limits the applicability of the conic Iris projection method. Also, the need for the system to operate with simple consumer device cameras restricts the use of infra-red lighting and stereoscopic imaging systems. As a result, the best fit in terms of accessibility and accuracy is the use of 3D model to compute the gaze vector. One aspect which makes up the complexity of imaging based gaze detection and tracking technologies which employ such a method is the fact that one cannot take into account the information about the subject's eyes without taking into account the information about the subject's face. Estimating the face orientation using only 4 key features is less complex, but is prone to significant estimation errors. Therefore it is necessary to extract and use as much information as possible from the face area in order to obtain a better approximation. This basically means tracking either the whole texture of the face or a considerable number of features.

Viola-Jones Object Detection: This is one of the most widespread methods used in face detection and tracking. It is based on training a cascade of small rectangular Haar-like feature detectors where each one is trained on the successful samples which passed its predecessor. Applying these simple classifiers on the integral image is done in constant time, therefore this algorithm is very attractive from the complexity and robustness standpoint [12]. Although this algorithm can only detect the position of the face and not the orientation, its low complexity makes it a suitable candidate as the first step in other more complex approaches. Unfortunately, the main drawback is the fact that the appearance of the face varies considerably with the head rotation and the deformations

which appear as a result of facial expressions and the classifier cascade can only detect faces similar with those it has been trained with (generally frontal faces). Although it would be possible to train more cascades, for various head orientation, the increase in running time incurred by the number of cascades necessary to support a sufficient orientation resolution would be prohibitive.

Active Shape Models: Used to detect and track various deformable objects, active shape models represent statistical models which are fitted to actual instances of images. They have applications in medical imaging and, more commonly, face detection. The shape is made up of a sequence of points in the image space, normally represented as a vector containing all the coordinates of the points. These points are chosen such that they overlap distinguishable features in the image, such as jaw lines, eye contours, mouth contours or eyebrows in the case of faces [13]. The model is built from a set of training images in which the landmark points are manually annotated. The average shape vector is computed and its main modes of variation are extracted using principal component analysis. These two components form the point distribution model, along with the image templates for each point make up the shape model. The main modes of variation coupled with additional parameters for global shape transformations such as rotation, translation, scaling and weak perspective projection represent the model parameters. The variations of the parameter values are constrained, thus allowing the shape to deform, but only as much and in such a way as seen in the training images [13]. An active shape model can be used both to fit a shape to a new image and to track an existing shape in a sequence of similar images such as video input. Both cases require an initial estimate of the parameters. When tracking, these estimates are trivially obtained from the previous frame. When fitting the shape to a new image, the global shape parameters can be obtained by running another algorithm such as Viola-Jones, and the deformation parameters are estimated to their mean values. The values of the model parameters are then iteratively refined. During each iteration, a new position is estimated for each landmark point nearby, according to their image templates. Then, the best variation of the model parameters is computed, such that, when applied, the landmark points are as close as possible to the new estimated positions, similarly to gradient descent [13].

Active Appearance Models: Active appearance models build upon active shape models by using all the data from the image which is inscribed in the shape boundary. This provides more accuracy in certain cases at the expense of running time, by taking into account more information which could be found in the image anyway. After the mean shape is computed, all training images are warped so that the landmark points are aligned with those in the mean shape, via triangulation. Now, all the appearance patches are aligned and the mean appearance and the main modes of variation can be computed, just like the shape. Thus, the model is enriched with new parameters corresponding to the appearance of the object. This approach enables the capability to actually generate new face textures within the model. This technique is used during the fitting process, by generating new estimates for the texture and then comparing it to the underlying image. These estimates are obtained by adjusting the current model parameter values by using the correlation between their values and the errors, which is learned in the off-line training phase [14]. Compared to the active shape models, active appearance models have the clear advantage of image generation capabilities and are more reliable because they make good use of the image intensity information. Therefore they could be used with a lower

number of landmark points, which make the annotation process of the training images considerably easier. On the other hand they are more computationally intensive and fail to generalize as well as active shape models on faces which are not part of the training set [15].

3. System Description

The gaze tracking system is made up of five main modules (Figure 1). The input image sequence is processed frame by frame. Each module handles a processing step. They are used sequentially to compute the gaze point for each frame. Every module except the first one processes the output data from the modules before it.

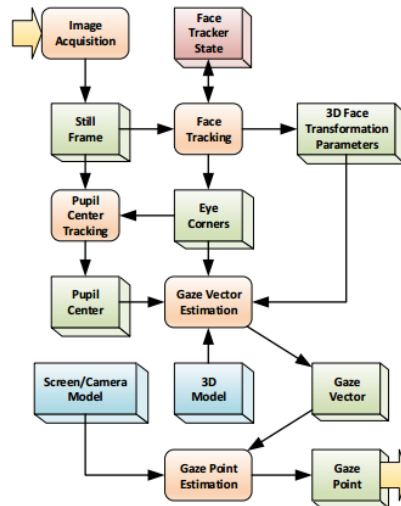


Figure 1. System architecture

The operations made by this system can be divided in two steps: (i) gaze tracking and (ii) detection of the pupil center. Each step is composed of several steps:

- gaze tracking:
 1. Poll the imaging device for the current frame.
 2. Fit the shape model to the face.
 3. Obtain face position, scale, orientation and eye corner coordinates from the shape model.
 4. Determine pupil centers.
 5. Compute the centers of the eyeballs.
 6. Compute eye gaze vector using the 3D model, face position, scale, orientation and the center of the eyeball.
 7. Intersect the gaze vector with the screen.
- Detection of the pupil center:
 1. Compute the midpoint of the line segment between the interior and exterior eye corners.
 2. Compute an affine transformation matrix such that the roll face rotation is inverted and the eye region of interest is scaled to a predefined size.
 3. Warp the image with this transformation matrix and obtain the eye region of interest.
 4. Convert the patch to grayscale.

5. Determine threshold such that the number of dark pixels is about a 10% of the number of the pixels in the patch.
6. Apply the inverse threshold filter with the determined value.
7. Get the coordinates of the non-zero pixels.
8. Compute the average coordinates.

The Image Acquisition Module: This module handles the capture of the input image sequence. It is required to poll the device's imaging device (i.e. camera/webcam) for the current frame. It operates in the 1st step of the gaze tracking algorithm. This module can also control the frame-rate of the whole system by introducing delays in polling. This feature is necessary because the system is computationally intensive, and because it should also be able to be used as a sub-system in another application. In the event that this application also requires a considerable amount of computing resources, it could dim necessary to use the gaze tracker at a smaller sample rate in order to use the remaining processing capabilities for additional, perhaps more complex tasks.

As the interaction with the capture device is platform dependent, the image acquisition module should also handle all the issues regarding this interaction, such as:

- If there are multiple imaging devices available, a decision has to be made about which one to use, based on available video modes. A balance needs to be established between the quality of the image and the increased complexity incurred by processing high-resolution images.
- As this module is the entry-point of the system, it is also responsible for reporting an error when no imaging device is available, or when connecting to available imaging devices fails. In this case the system will gracefully transition to an inactive state.
- In case of a temporary malfunction, such as failing to poll the imaging device for the current frame, this module reports an error. The system remains in an active state, so that in the event of a remedy it will continue to function properly.

This module does not require any input data because it is the integration point between the gaze tracking system and the hardware. The output of this module is represented by the image data of the current frame. This is passed to the face tracking and the pupil center tracking modules.

The Face Tracking Module: This module implements the active shape model fitting and tracking algorithms. It operates on both the 2nd and 3rd steps of the gaze tracking step. Its responsibilities include:

- Loading the statistical model from file.
- Keeping track of the status of the previous frame (whether the face was tracked successfully or not). - Reinitialize the tracker in the case of a shape fitting failure in the previous frame.
- Track the shape of the face based on a successful fit in the previous frame.

The input of this module comes from the image acquisition module in the form of the current frame and from the internal state manager which provides the previous frame image data and the shape parameters.

If the face could not be detected, no output is provided and this error state is signaled by this module. If the shape model is successfully fitted to the current frame, then the output consists of two main components:

- The 3D face transformation parameters which include:

- The X and Y translation offsets in the image plane, defined as the pixel differences of the coordinates of the center (the mean of the landmark coordinates) of the face shape and the upper-left corner of the image.
- The scaling factor of the face, relative to an arbitrary, but consistent baseline.
- The Euler angles (i.e. roll, pitch, and yaw) of the face, with a 0° reference for a straight frontal face.
- The face normal vector will also be a part of the face tracking module's output, for convenience.
- The coordinates of all the points which make up the landmark set of the face model, specifically the four points which represent the eye corners.

The Pupil Center Tracking Module: This module is used twice per frame, once for each eye. It handles the 4th step of the gaze tracking step and, internally, it works according the detection of the pupil center step. An exception during its invocation occurs when one of the eyes is occluded. The input of this module consists of the output data of the two modules before it in the processing chain:

- The coordinates of the eye corners from the face tracker are required in order to define a narrow window around the eye in which to search for the pupil center. This narrow search windows speeds up the process which would otherwise require to search the whole image for the two irises.
- The actual image data for the current frame from the image acquisition module is required in order to extract the image intensity in the regions of interest.

The output of this module consists of the X and Y coordinates of the center of the pupil, measured in pixel units from the top left corner of the image.

The Gaze Vector Estimation Module: This module performs the computation of the three-dimensional gaze vector and is needed at the 5th and 6th steps in the gaze tracking step. This is where the transition from the two-dimensional space of the input image stream to the three-dimensional space extrapolated using the 3D model is performed. This model is loaded during the module's initialization phase. This is also the module responsible for most of the final approximation error as it aggregates data from different processing paths. This is the reason why this module could benefit from using an additional non-linear classifier such as a back-propagation artificial neural network to correct these errors. The input consists of four main components:

- The face affine transformation parameters from the face tracking module. These are used to position the subject's face in the extrapolated 3D space.
- The locations of the eye corner landmark points from the fitted shape. These are also transposed into the 3D space and are required in order to determine the position of the eyeball centers.
- The positions of the pupil centers from the pupil tracking module are also transposed in the 3D space using the 3D model. Here, along with the eyeball center, they determine the gaze vector.
- The 3D model is a static set of anatomical constants and scaling factors used to extrapolate the depth of certain locations in the 2D input image.

The Gaze Point Estimation Module: This final module in the frame processing path handles the 7th step from gaze tracking step. Its responsibility is to compute the intersection of the gaze vector with the device's screen. Depending of the system's use

cases, this module could be suppressed so that the final output is the gaze vector. More complex custom computations could be performed using the additional data encoded in a 3D vector, compared to a 2D point. The required input is made up of the gaze vector from the output of the previous module and the camera/screen 3D model. This model is device dependent and can be setup using an initial calibration procedure before the first use of the system on a certain device. This model should take into account:

- The width and height of the screen.
- The spatial resolution of the imaging device.
- The lens distortions (if they are significant).
- The imaging device field of view.
- The relative position of the camera and the screen.

The output of this module consists of a single 2D point corresponding to the subject's point of gaze on the screen. This is measured in pixel units from the top-left corner of the screen.

4. Experimental Results

The system was tested on an ASUS N65JV notebook computer running OpenSUSE Linux 13.1, with a dual core Intel i5 430M CPU, clocked at 2.2GHz and 4GB of RAM memory. The imaging device is the integrated web camera, which provides 10 frames per second at VGA (640x480 pixel) resolution. The gaze point was tracked on the 16'' 1366x768 pixel display (35x20 cm). On this hardware configuration, the gaze tracking system runs at 35Hz using version 2.4.8 of the OpenCV library.

Figure 2, 3, 4 and 5 plot the estimation error. This error is defined as the angle between the real gaze vector and the estimated one. In order to provide a value for the actual gaze vector, the following testing scenario is followed: a user is performing head movements consistent with each of the measurement test cases. During all the time the data is recorded, she keeps fixating at a predetermined point on the screen. As the gaze vector corresponding to this point is known, the difference between it and the estimation obtained using the gaze tracking system is computed.

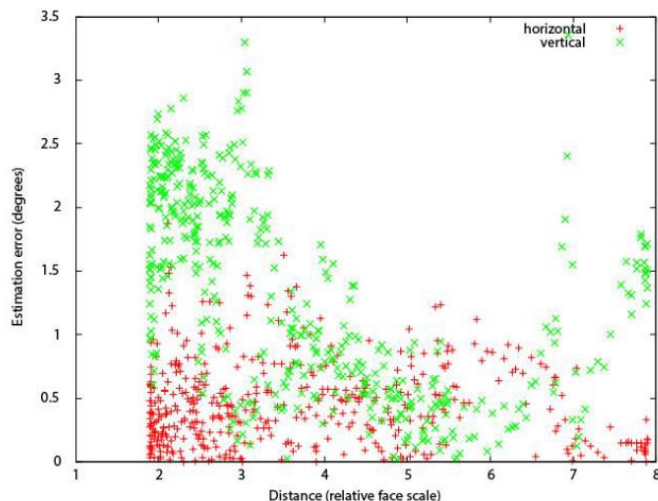


Figure 2. Estimation error by distance between the face and the camera

The subject's head performs a translation motion away from the camera. For a more explicit analysis of the system, the error is split into two components: horizontal and vertical. The red points represent the horizontal absolute value of the error, across the screen width. The green points represent the horizontal absolute value of the error, across the screen height.

Figure 2 shows the relationship between how far away from the camera a user is situated and the gaze tracker error. The distance on the graph is represented as the relative scale of the face, obtained from the face tracker. This is inversely proportional to the distance. Approximate real world values can be associated with measurements of 75cm at a 2.5 relative scale and 27cm at a 7.0 relative scale. The measurements in Figure 2 are taken in the constrained scenario where the user is allowed only a translation movement along the axis normal to the screen which passes through its middle. This means that this baseline measurement was obtained for frontal face use cases. The plotted data illustrates that the average estimation error is greater as the user moves further away from the camera. This is the expected behavior, as the resolution of the eye region of interest decreases linearly with regard to the face scale.

The subject's head a variety of rotations and translation movements, similar to the natural motion of a user sitting at a computer. It is also noticeable that the absolute value of the slope of the average estimation error is greater vertically than horizontally. This is another expected result, explained by the shape of the human eye lids which open to uncover a region of the eye which is larger in width rather than in height. Thus, the visible sclera above and below the iris has a smaller surface area than the visible sclera to the left and to the right of the iris. In certain cases, the sclera might not be visible at all above and below the iris. This causes in fact a sampling resolution for the center of the iris which is horizontally greater than vertically.

The accuracy also suffers when the face scale has a value around 7 or higher. This is caused by the fact that parts of the face are positioned outside the frame and the face tracker does not handle this case appropriately. This behavior could benefit from a future development, either by considering that the face is not tracked in these conditions and ignoring the following steps completely, or by adding virtual borders to the input frame and treat this case the same as a face occlusion.

Figure 3 shows the samples taken while the user is allowed full head motion, with the single constraint of staring at a fixed point. As expected, the absolute values of the estimation errors are higher than those in the scenario from Figure 2, because additional variables have to be taken into account by the gaze tracker, such as the orientation of the head and the offset of the face inside the frame. All these variable introduce additional errors. Nonetheless, the same trend can be observed, with the vertical error being higher than the horizontal error on average while both datasets present higher error values for smaller values of the face scale.

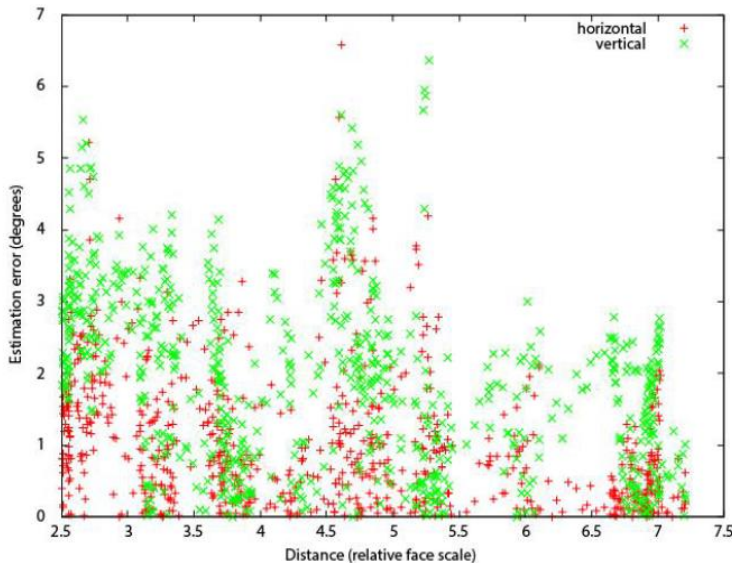


Figure 3. Estimation error by distance between the face and the camera with normal movement

The center of the subject's head is kept at an approximately fixed position while a 30° yaw rotation is performed (left to right).

Figure 4 presents the gaze estimation error as a function of the yaw angle of the subject's head. As the head performs a rotation on the ground vertical axis, the error is shown to increase as the absolute value of the angle increases. The lower values of the error measurements compared to the free head motion test case are explained by the positioning of the subject head at a closer distance to the camera, corresponding to a face scale of 5. This decreases the contribution of the low resolution captured frames towards these measured error values.

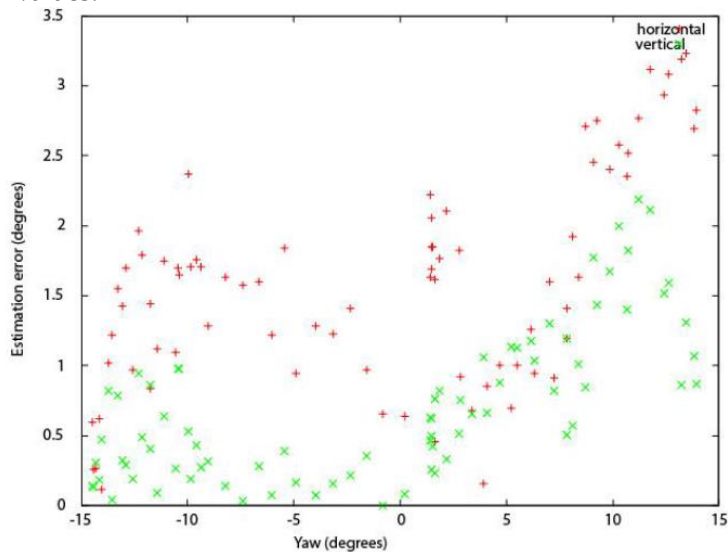


Figure 4. Estimation error by head yaw rotation

As the eyeballs maintain their vertical orientation in this scenario, it is noticeable that the vertical estimation error is lower than the horizontal one. The greater values for the horizontal values stem mainly from the simplifications used in the 3D model. These simplifications introduce new sources of approximation errors while computing the center of the eyeball. The center of the subject's head is kept at an approximately fixed position while a 45° pitch rotation is performed (from facing up to facing down).

In Figure 5 the estimation error is plotted as a function of the pitch angle of the user's head. This scenario involves the subject keeping her gaze at a fixed point on screen plane while changing the pitch of the head from 15° facing upwards to 30° facing downwards. The overall accuracy is still better at 0° (i.e. with a frontal face), just like when varying the yaw. What's different in this case is the divergence of the two components.

When facing up (and gazing down upon the screen), the upper eyelid involuntarily covers the most of the upper part of the eyeball. This results in a very small opening between the two eyelids through which the sclera and the iris are observed. This is only slightly wider than the pupil diameter and is furthermore obstructed by the eye lashes. In this configuration, the center of the pupil lies roughly on the axis between the two corners of the eye. Therefore, when searching for the darkest pixels in the image, a reasonable approximation is obtained for its vertical position. The same hold true for the horizontal position of the center of the pupil, provided that the sclera is not completely covered so that a sufficient contrast level between it and the darker pupil/iris is captured in the frame. As a result, the errors are roughly the same for both horizontal and the vertical estimation, with values larger than those corresponding to a frontal face because of the smaller observable region of interest.

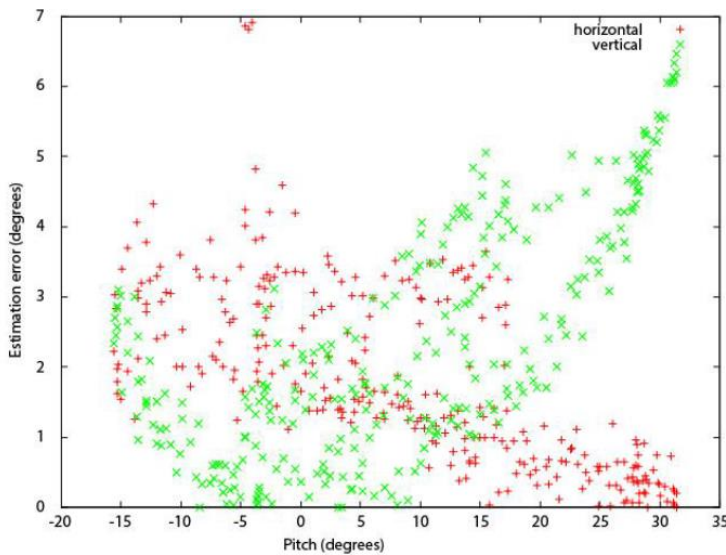


Figure 5. Estimation error by head pitch rotation

On the opposite end of the pitch angle range, when facing down and looking upwards at the screen, the subject's eyes are wide open. Because the rotation of the eyeballs around the vertical axis does not change in this scenario, coupled with the improved size of the area of interest in the frame, the horizontal accuracy improves considerably.

On the other hand, the vertical estimation of the gaze angle increases very rapidly as the head is tilted further down. The root cause of this loss of accuracy is the method for detecting the iris center. When the head is maximally tilted down and the user is looking forward, the eyeballs are rotated upward at the maximum possible angle relative to the eye socket. This cause almost the full upper half of the iris to fall below the upper eyelid. Therefore, in this case, the average location of the darkest pixels is no longer as close to the center of the pupil as in the case when the whole iris and the pupil are visible. It is instead slightly lower, therefore the error in the vertical gaze angle estimation increases.

5. Conclusions and Future Work

This paper describes a real-time and non-intrusive system for gaze tracking. By using a single monocular low-resolution imaging device, it was possible to extract from the frame stream information such as the face orientation, scale and position, the centers of the eyeballs and the centers of the irises. Using this information, together with the three-dimensional head model and the anatomical constants which were determined experimentally, the system computes the gaze vector of each of the eyes. The average of these vectors is computed in order to minimize the noise affecting each individual measurement. The final step required for a variety of modern HCI applications is the computation of the gaze point of the user. By computing the location of the gaze point on the device display, the gaze tracking system provides an additional control device for digital equipment and attempts to weaken the barrier between humans and machines. Despite its inferior accuracy, this gaze tracker can still be integrated into other systems to enable new, interesting and creative ways for the users to interact with certain applications. As future work we include additional parameters in the 3D model: depending on the exact definition of the locations of the corners of the eye, the center of the eyeball might not lie exactly on the axis passing through the midpoint of the segment defined by the two corners, which is parallel to the normal to the face plane. In order to mitigate this erroneous assumption, the 3D model could include a displacement for the actual location of this point, on the X and Y axes in the head reference system.

Acknowledge: The research presented in this paper is co-funded by the national project Mobile@Old, PN-II-PT-PCCA-2013-4-2241 No 315/2014, under the Partnerships Program PN II, powered by MEN – UEFISCDI.

REFERENCES:

- [1] J. Magee, M. Betke, J. Gips, M. Scott and B. Waber, "A Human-Computer Interface Using Symmetry Between Eyes to Detect Gaze Direction," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 6, pp. 1248-1261, November 2008.
- [2] J. Najemnik and W. Geisler, "Optimal eye movement strategies in visual search," *Nature*, pp. 387-391, 17 March 2005.
- [3] T. Ishikawa, S. Baker, I. Matthews and T. Kanade, "Passive Driver Gaze Tracking with Active Appearance Models," in *The 11th World Congress on Intelligent Transportation Systems*, Nagoya, Japan, 2014.
- [4] Q. Ji, Z. Zhu and P. Lan, "Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue," *Vehicular Technology, IEEE Transactions on*, vol. 53, no. 4, pp. 1052-1068, 2004.

- [5] B. Blaus, "Anatomy of the Eye," Wikimedia Commons, 2013 October 2013. [Online]. Available: http://commons.wikimedia.org/wiki/File:Blausen_0388_EyeAnatomy_01.png. [Accessed April 2015].
- [6] A. H. Gee and R. Cipolla, "Determining The Gaze Of Faces In Images," *Image and Vision Computing*, vol. 12, pp. 639-647, 1994.
- [7] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, Grenoble, France, 2000.
- [8] E. Pogalin, A. Redert, I. Patras and E. A. Hendriks, "Gaze Tracking by Using Factorized Likelihoods Particle Filtering and Stereo Vision," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, Chapel Hill, USA, 2006.
- [9] J. Qiang, Z. Zhiwei and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *Vehicular Technology, IEEE Transactions on*, vol. 53, no. 4, pp. 1052-1068, July 2004.
- [10] W. Haiyuan, C. Qian and T. Wada, "Conic-based algorithm for visual line estimation from one image," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, Seoul, Korea, 2004.
- [11] W. Zhang, T. Zhang and S. Chang, "Gazing Estimation and Correction from Elliptical Features of One Iris," in *Image and Signal Processing (CISP), 2010 3rd International Congress on*, Yantai, China, 2010.
- [12] P. Viola and M. Jones, "Robust Real-Time object detection," *International Journal of Computer Vision*, vol. 4, pp. 34-47, 2001.
- [13] T. F. Cootes and C. J. Taylor, "Active Shape Models - 'Smart Snakes'," in *British Machine Vision Conference*, Leeds, United Kingdom, 1992.
- [14] G. J. Edwards, C. J. Taylor and T. F. Cootes, "Interpreting face images using active appearance models," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, Nara, Japan, 1998.
- [15] T. F. Cootes, G. J. Edwards and C. J. Taylor, "Comparing Active Shape Models with Active Appearance Models," in *Tenth British Machine Vision Conference*, Nottingham, United Kingdom, 1999.