Simple and Efficient Visual Gaze Estimation

Roberto Valenti
r.valenti@uva.nl

Nicu Sebe
n.sebe@uva.nl
Intelligent Systems Lab
Amsterdam
Kruislaan 403, 1018SJ
Amsterdam, The Netherlands

Theo Gevers
th.gevers@uva.nl

ABSTRACT

In this paper we combine a state of the art eye center locator and a new eye corner locator into a system which estimates the visual gaze of a user in a controlled environment (e.g. sitting in front of a screen). In order to reduce to a minimum the computational costs, the eye corner locator is built upon the same technology of the eye center locator, tweaked for the specific task. If high mapping precision is not a priority of the application, we claim that the system can achieve acceptable accuracy without the requirements of additional dedicated hardware. We believe that this could bring new gaze based methodologies for human-computer interactions into the mainstream.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications; J.7 [**Computers in Other Systems**]: Command and control

General Terms

Algorithms

Keywords

Eye tracking, ambient intelligence, visual gaze, HCI

1. INTRODUCTION

Eye location and tracking and the related visual gaze estimation are important tasks in many computer vision applications and research [1]. Some of the most common examples are the application to user attention and gaze in driving and marketing scenarios, and control devices for disabled people. Eye location/tracking techniques can be divided into three distinct modalities [10]: (1) Electro oculography, which records the electric potential differences of the skin surrounding the ocular cavity; (2) scleral contact lens/search coil, which uses a mechanical reference mounted on a contact lens, and (3) photo/video oculography, which uses image processing techniques to locate the center of the eye. Unfortunately, the common problem of the above techniques is the use of intrusive and expensive sensors [4]. While photo/video oculography is considered the least invasive of the modalities, commercially available trackers still require the user to be either equipped with a head mounted device, or to use a high resolution camera combined with a chinrest to limit the allowed head movement. Furthermore, daylight applications are precluded due to the common use of active infrared (IR) illumination, used to obtain accurate eye location through corneal reflection. Non infrared appearance based eye locators [2, 3, 5, 7, 8, 12, 11, 15, 19] can successfully locate eye regions, yet are unable to track eye movements accurately.

The goal of this paper is to build an eye tracker that can quickly and accurately locate and track eye centers and eye corners in low resolution images and videos (coming from a simple web cam) and map them on the screen plane. The rest of this paper is structured as follows: For completeness, the theory behind the used framework is explained in section 2 (for additional details refer to [16]). Section 3 describes how to accurately detect an eye center starting from an image of an eye, Section 4 applies the same rationale to eye corner location, with the help of some simple geometrical constraints. Finally, in section 5 we will discuss how to easily map eye and corner location to screen coordinates.

2. ISOCENTERS ESTIMATION

The isophotes of an image are curves connecting points of equal intensity. Since isophotes do not intersect each other, an image can be fully described by its isophotes. Furthermore, the shape of the isophotes is independent to rotation and linear lighting changes [14]. To better illustrate the well known isophote framework, it is opportune to introduce the notion of intrinsic geometry, geometry with a locally defined coordinate system. In every point of the image, a local coordinate frame is fixed in such a way that it points in the direction of the maximal change of the intensity, which corresponds to the direction of the gradient. This reference frame $\{v, w\}$ is referred to as the *gauge coordinates*. Its frame vectors $\hat{w}$ and $\hat{v}$ are defined as:

$$\hat{w} = \frac{\{L_x, L_y\}}{\sqrt{L_x^2 + L_y^2}}; \hat{v} = \perp\hat{w}; \qquad (1)$$

where L_x and L_y are the first-order derivatives of the luminance function $L(x, y)$ in the x and y dimension, respectively. In this setting, a derivative in the w direction is the gradient itself, and the derivative in the v direction (perpendicular to the gradient) is 0 (no intensity change along the isophote). In this coordinate system, an isophote is defined as $L(v, w(v)) = constant$ and its curvature κ is defined as the change w'' of the tangent vector w' which in Cartesian coordinates becomes [9, 17, 16]

$$\kappa = -\frac{L_{vv}}{L_w} = -\frac{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}}. \qquad (2)$$

Since the curvature is the reciprocal of the radius, we can reverse Eq. (2) to obtain the radius of the circle that generated the curvature of the isophote. The radius is meaningless if it is not combined with orientation and direction. The orientation can be estimated from the gradient, but its direction will always point towards the highest change in the luminance. However, the sign of the isophote curvature depends on the intensity of the outer side of the curve (for a brighter outer side the sign is positive). Thus, by multiplying the gradient with the inverse of the isophote curvature, the duality of the isophote curvature helps in disambiguating the direction of the center. Since the gradient can be written as $\frac{\{L_x, L_y\}}{L_w}$, we have

$$\begin{aligned} D(x,y) &= \frac{\{L_x, L_y\}}{L_w}\left(-\frac{L_w}{L_{vv}}\right) = -\frac{\{L_x, L_y\}}{L_{vv}} \\ &= -\frac{\{L_x, L_y\}(L_x^2 + L_y^2)}{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}. \qquad (3) \end{aligned}$$

where $D(x, y)$ are the displacement vectors to the estimated position of the centers, which can be mapped into an accumulator, hereinafter *"centermap"*. Since every vector gives a rough estimate of the center, we can convolve the accumulator with a Gaussian kernel so that each cluster of votes will form a single center estimate. Furthermore, the contribution of each vector can be weighted according to a relevance mechanism. The main idea is that by collecting and averaging local evidence of curvature, the discretization problems in a digital image could be lessened and accurate center estimation could be achieved.

In order to achieve this goal, only the parts of the isophotes which are meaningful for our purposes should be used, that is, the ones that follow the edges of an object. This selection can be performed by using the curvedness [13]:

$$curvedness = \sqrt{L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2}. \qquad (4)$$

We note that the curvedness has low response on flat surfaces and edges, whereas it yields high response in places where the isophote density is maximal. As observed before, the isophote density is maximal around the edges of an object, meaning that by selecting the parts of the isophotes where the curvedness is maximal, they will likely follow an object boundary and locally agree on the same center. The advantage of this approach over a pure edge based method is that, by using the curvedness as the voting scheme for the importance of the vote, every pixel in the image may contribute to a decision. By summing the votes, we obtain high response on isocentric isophotes patterns which respect the constraint of being near edges. We call these high responses *"isocenters"*, or ICs.

3. EYE CENTER LOCATION

Recalling that the sign of the isophote curvature depends on the intensity of the outer side of the curve, we observe that a negative sign indicates a change in the direction of the gradient (from brighter to darker areas). Therefore, it is possible to discriminate between dark and bright centers by analyzing the sign of the curvature. Regarding the specific task of cornea and iris location, it can be assumed that the sclera is brighter than the cornea and the iris, so we should ignore the votes in which the curvature is positive, that is, where it agrees with the direction of the gradient. As a consequence, the maximum isocenter (MIC) obtained will represent the estimated center of the eye. The result of this procedure on an eye image is shown in figure 1. From the 3D plot it is clear where the MIC is, but we can expect that certain lighting conditions and occlusions from the eyelids to result in a wrong eye center estimate. To cope with this problem, we use the mean shift algorithm for density estimation. Mean shift (MS) usually operates on back-projected images in which probabilities are assigned to pixels based on the color probability distribution of a target, weighted by a spatial kernel over pixel locations. It then finds the local maximum of this distribution by gradient ascent [6]. Here, the mean shift procedure is directly applied to the centermap resulting from our method, under the assumption that the most relevant isocenter should have higher density of votes, and that wrong MICs are not so distant from the correct one (on an eye corner). A mean shift search window is initialized on the centermap, centered on the found MIC. The algorithm then iterates to converge to a region with maximal votes distribution. After some iteration, the isocenter closest to the center of the search window is selected as the new eye center estimate.

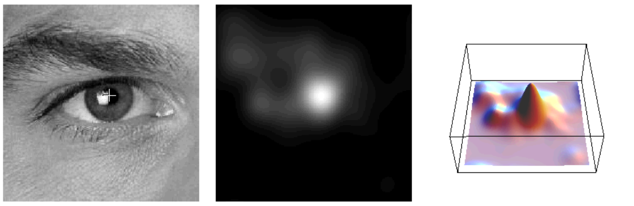

Figure 1: The source image, the obtained centermap and the 3D representation of the latter

An extensive evaluation of the eye locator was performed in [16], testing the eye locator for robustness to illumination and pose changes, for accurate eye location in low resolution images and for eye tracking in low resolution videos. The comparison with the state of the art suggested that the method is able to achieve highest accuracy, but this is somewhat bounded by the presence of a symmetrical pattern in the image. Figure 2 qualitatively shows some of the results obtained on different subjects of the BioID database. The dataset consists of 1521 grayscale images of 23 different subjects and has been taken in different locations and at different times of the day (uncontrolled illumination). We observe that the method successfully deals with slight changes in pose, scale, and presence of glasses (second row). By analyzing the failures (last row) it can be observed that the system is prone to errors when presented with closed eyes, very bright eyes, or strong highlights on the glasses. When these cases occur, the iris and cornea do not contribute enough to the center voting, so the eyebrows or the eye corners assume a position of maximum relevance.

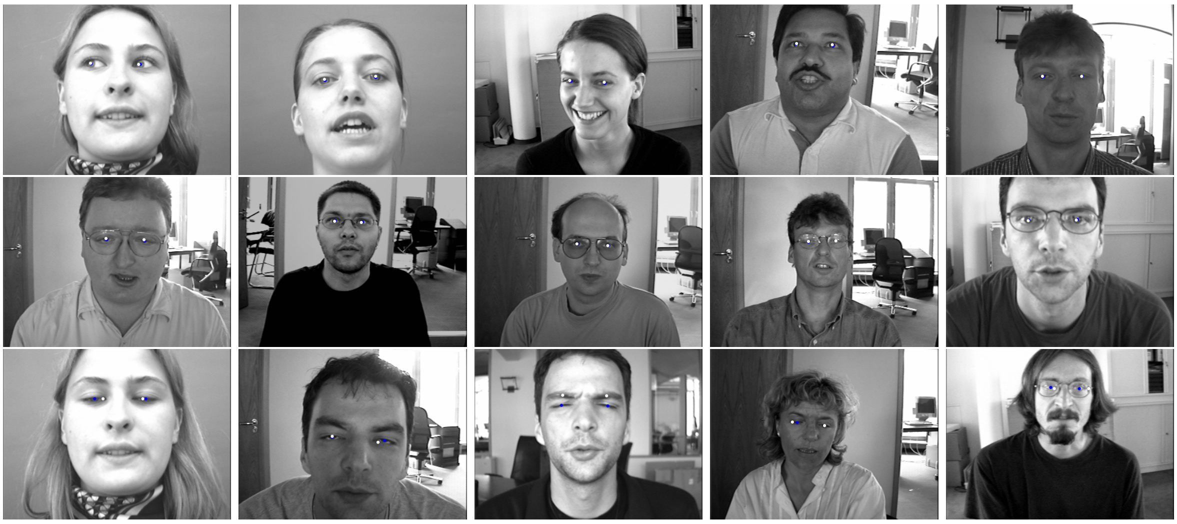

Figure 2: Sample of success and failures (last row) on the BioID face database; a white dot represents the estimated center.

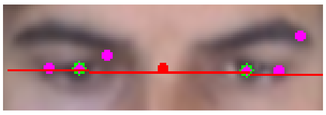

Figure 4: Eye centers and corner candidates

4. EYE CORNER LOCATION

Unfortunately the eye center location is not enough for visual gaze estimation: there is a need for an accurate fixed point (or anchor point) in order to be able to measure successive displacements of the eye center independently of the face position. The common approach is to locate the position of the eyelids and the eye corners [18, 20]. A fast and inexpensive way to locate such an anchor is to reuse the obtained centermap. As stated before, by analyzing the results of the eye locator we note that the largest number of mistakes in eye-center location are located on eye corners. This is due to the fact that the eye corners have a somewhat symmetrical structure: in blurred low resolution imagery, the junction between the eyelashes creates an almost circular dark structure which is in contrast with the brighter skin and the sclera and therefore receives higher response than the rest of the features. In this way we can exploit this problem to our advantage. Figure 4 shows the highest ICs obtained. Once the eye center is selected by the mean shift we can apply some geometrical constraints to find the most stable anchor. Experimentally, the external eye corner turned out to be the most stable isocenter. In order to find them we look for the furthest away isocenter that lays closer to the line created by connecting the two eye centers (shown in red in figure 4). While this assumption is reasonable and showed quite stable results (see figure 3), the process is bound to fail every time that the eye locator fails (last image in figure 3). This problem could be solved by enforcing additional constrains on the movement.

5. VISUAL GAZE ESTIMATION

Now that we have the eye center and corner location available, in order to correctly estimate visual gaze it would be reasonable to consider the head position and orientation to give a rough initialization of the visual gaze, and then use the information about the eye centers and corners to fine tune the information. Unfortunately, head pose estimators often involve many assumptions in order to achieve a realistic modeling (i.e. the shape and size of the head, the possible rotation angles of the eye, etc.). Furthermore, the high computational requirements of head pose estimators are not in line with the lightweight requirements of our system. Finally small mistakes in pose estimation might introduce additional errors in the final visual gaze estimation. Other methods tend to simplify the problem by assuming that the eye doesn't rotate but it just shifts. This assumption is reflected in commercial eye trackers, which deal with high resolution images of the eyes. This simplification comes from the assumption that the face is always frontal to the screen so the head pose information can be discarded. Therefore we used the linear mapping method suggested by [20], the user needs to perform a calibration procedure by looking at several known points on the screen. A 2D linear mapping is then constructed from the vector between the eye corner and the iris center and recorded at the known position on the screen. This vector is then used to interpolate between the known screen locations. For example, if we have two calibration points P_1 and P_2 with screen coordinates α and β, and eye-center vector (taken with origin from the anchor point) x and y, we can interpolate a new reading of the eye-center vector to obtain the screen coordinates by using the following interpolant:

$$\alpha = \alpha_1 + \frac{x - x_1}{x_2 - x_1}(\alpha_2 - \alpha_1) \tag{5}$$

$$\beta = \beta_1 + \frac{y - y_1}{y_2 - y_1}(\beta_2 - \beta_1) \tag{6}$$

The advantage of this approach is its low computational cost and a decent accuracy with respect to more complex systems. In facts the reported error introduced by this approximation is just 1.2°. Unfortunately, this method doesn't allow head movements, so the user will need to recalibrate in case of big horizontal or vertical shifts. However, if the dis-

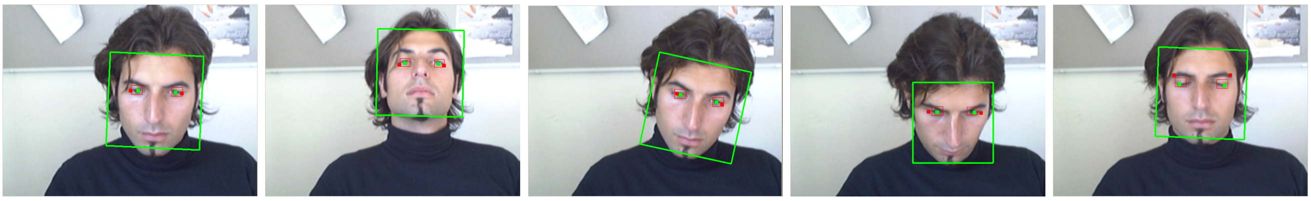

Figure 3: Examples of combined eye center (green) and eye corner (red) detection

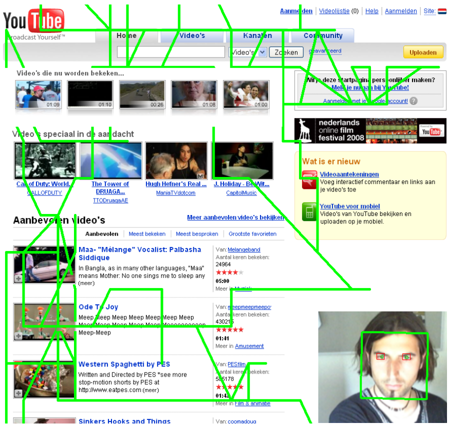
YouTube
Broadcast Yourself™
Aanmelden | Videolijstje (0) | Help | Aanmelden | Site:
Home
Video's
Kanalen
Community
Zoeken
Uploaden
Video's die nu worden bekeken...
Video's speciaal in de aandacht
Wil je deze startpagina persoonlijker maken? Meld je nu aan bij YouTube!
nederlands online film festival 2008
Wat is er nieuw
Videoaantekeningen
YouTube voor mobiel
Aanbevolen video's
Meer aanbevolen video's bekijken
Maa- "Mélange" Vocalist: Palbasha Siddique
Ode To Joy
Western Spaghetti by PES
Sinkers Hooks and Things

Figure 5: Mapped visual gaze on an internet page

tance from the screen and the camera parameters are known it is also possible to compensate for this problem by remapping the calibration points accordingly with the registered displacement of the eyes. Therefore the final accuracy of the system is bounded just by the number of pixels that the eye is allowed to move. This generates some kind of *grid effect* on the recorded eye locations that can be seen in figure 5. While the final accuracy is bounded by the quality of the camera and the distance from it, we still believe that the system can be used for specific applications that don't require high level of accuracy (like changing the focused window or scrolling when looking outside the boundaries of the screen).

6. CONCLUSIONS

In this paper, we extended a method to infer eye center location to eye corner detection. Both eye center and eye corner can be detected at same time, not requiring significant additional computation. We used the estimated locations to estimate the visual gaze of a user sitting in front of a screen. Although the accuracy of the system is bounded by the quality of the used webcam, we believe that the approximate gaze information can be used to deeply enhance human-computer interaction.

7. REFERENCES

[1] COGAIN communication by gaze interaction, gazing into the future. http://www.cogain.org, 2006.

[2] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas. An eye detection algorithm using pixel to edge information. In *Int. Symp. on Control, Commun. and Sign. Proc.*, 2006.

[3] L. Bai, L. Shen, and Y. Wang. A novel eye location algorithm based on radial symmetry transform. In *ICPR*, pages 511–514, 2006.

[4] R. Bates, H. Istance, L. Oosthuizen, and P. Majaranta. Survey of de-facto standards in eye tracking. In *COGAIN Conf. on Comm. by Gaze Inter.*, 2005.

[5] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization through a general-to-specific model definition. In *BMVC*, 2006.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–577, 2003.

[7] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006.

[8] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In *BMVC*, pages 277–286, 2004.

[9] E. B. Dam and B. ter Haar Romeny. *Front End Vision and Multi-Scale Image Analysis*. Kluwer, 2003.

[10] A. T. Duchowski. *Eye Tracking Methodology: Theory and Practice*. Springer, 2007.

[11] M. Hamouz, J. Kittlerand, J. K. Kamarainen, P. Paalanen, H. Kalviainen, and J. Matas. Feature-based affine-invariant localization of faces. *PAMI*, 27(9):1490–1495, 2005.

[12] O. Jesorsky, K. J. Kirchbergand, and R. Frischholz. Robust face detection using the Hausdorff distance. In *Audio and Video Biom. Pers. Auth.*, pages 90–95, 1992.

[13] J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, pages 557–565, 1992.

[14] J. Lichtenauer, E. Hendriks, and M. Reinders. Isophote properties as features for object detection. In *CVPR*, volume 2, pages 649–654, 2005.

[15] M. Türkan, M. Pardás, and A. Çetin. Human eye localization using edge projection. In *Comp. Vis. Theory and App.*, 2007.

[16] R. Valenti and T. Gevers. Accurate eye center location and tracking using isophote curvature. In *CVPR*, June 2008.

[17] M. van Ginkel, J. van de Weijer, L. van Vliet, and P. Verbeek. Curvature estimation from orientation fields. In *SCIA*, 1999.

[18] Z. Zheng, J. Yang, and L. Yang. A robust method for eye features extraction on color image. In *Pattern Recognition Letters*, volume 26, pages 2252–2261, 2005.

[19] Z. H. Zhou and X. Geng. Projection functions for eye detection. In *Pattern Recog.*, pages 1049–1056, 2004.

[20] J. Zhu and J. Yang. Subpixel eye gaze tracking. In *FGR*, page 131. IEEE Computer Society, 2002.