Accurate Eye Center Location and Tracking Using Isophote Curvature

Roberto Valenti* Theo Gevers
Intelligent Systems Lab Amsterdam
University of Amsterdam, The Netherlands
{rvalenti,gevers}@science.uva.nl

Abstract

*The ubiquitous application of eye tracking is precluded by the requirement of dedicated and expensive hardware, such as infrared high definition cameras. Therefore, systems based solely on appearance (*i.e.* not involving active infrared illumination) are being proposed in literature. However, although these systems are able to successfully locate eyes, their accuracy is significantly lower than commercial eye tracking devices. Our aim is to perform very accurate eye center location and tracking, using a simple web cam. By means of a novel relevance mechanism, the proposed method makes use of isophote properties to gain invariance to linear lighting changes (contrast and brightness), to achieve rotational invariance and to keep low computational costs. In this paper we test our approach for accurate eye location and robustness to changes in illumination and pose, using the BioID and the Yale Face B databases, respectively. We demonstrate that our system can achieve a considerable improvement in accuracy over state of the art techniques.*

1. Introduction

Eye location and tracking are important tasks in many computer vision applications and research [7]. Some of the most common examples are the application to face alignment, face recognition, user attention and gaze (*e.g.* driving and marketing), and control devices for disabled people. Eye location/tracking techniques can be divided into three distinct modalities [12]: (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 [3]. 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 [1, 2, 5, 9, 10, 16, 15, 22, 25] (Section 4.3) can successfully locate eye regions, yet are unable to track eye movements accurately. Approaches that fuse IR and appearance based modalities are also proposed in literature [26], but dedicated hardware is still required.

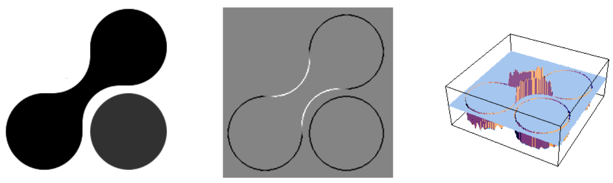
The goal of this paper is to build an eye detector and tracker that can quickly and accurately locate and track eye centers in low resolution images and videos (*i.e.* coming from a simple web cam). For this we made the following contributions: (1) A novel eye location approach is proposed, which is based on the observation that eyes are characterized by radially symmetric brightness patterns. Contrary to other approaches using symmetry to accomplish the same task [2], our method uses isophotes (Section 2) to infer the center of (semi)circular patterns. (2) A novel center voting mechanism (introduced in Section 3) which is the key for the successful outcome of our approach. This mechanism is used to increase and weight important votes to reinforce the center estimates. In this paper we study the accuracy and the robustness of the proposed approach to lighting and pose changes, and compare the obtained results with the state of the art systems for eye location in low resolution imagery (Section 4).

2. Isophotes Curvature Estimation

The isophotes of an image are curves connecting points of equal intensity (one could think of isophotes as contour lines obtained by slicing the intensity landscape with horizontal planes). 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 [19]. Due to these properties, isophotes have been successfully used as features in object detection and image segmentation [13, 17, 19]. To better illustrate the well known isophote framework, it is opportune

*Supported by the MIAUCE European Project (IST-5-0033715-STP).

978-1-4244-2243-2/08/$25.00 ©2008 IEEE



(a) (b) (c)

Figure 1. The original image (a), its isophote curvature at the edges (b), and the 3D plot of the latter (c)

to introduce the notion of intrinsic geometry, *i.e.* 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:

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

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 is defined as the change w'' of the tangent vector w'. By implicit differentiation with respect to v of the isophote definition, we have:

$$L_v + L_w w' = 0; \qquad w' = -\frac{L_v}{L_w}. \tag{2}$$

Since we know from the gauge condition that $L_v = 0$, we have $w' = 0$. Differentiating again with respect to v, yields

$$L_{vv} + 2L_{vw}w' + L_{ww}w'^2 + L_w w'' = 0. \tag{3}$$

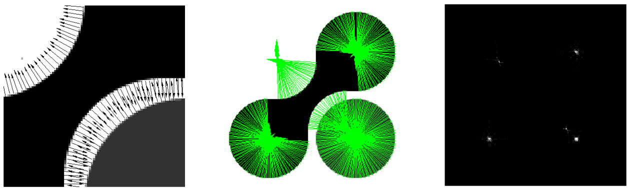
Solving for $\kappa = w''$ (the isophote curvature) and knowing that $w' = 0$, the isophote curvature is obtained as

$$\kappa = -\frac{L_{vv}}{L_w}. \tag{4}$$

In Cartesian coordinates, this becomes [11, 23]

$$\kappa = -\frac{L_{vv}}{L_w} = -\frac{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}}. \tag{5}$$

The isophote curvature of a toy image is shown in Figure 1(b). For presentation purposes, the shown curvature belongs to the isophote under the edges found in the image using a Canny operator. The crown-like shape of the values in the 3D representation (Figure 1(c)) is generated by the aliasing effects on the circle. By scaling[1] the original image this effect is reduced, but at higher scales the isophotes curvature might degenerate with the inherent effect of losing important structures in the image.



(a) (b) (c)

Figure 2. The direction of the gradient under the image's edges (a), the displacement vectors pointing to the isophote centers (b), and the centermap (c)

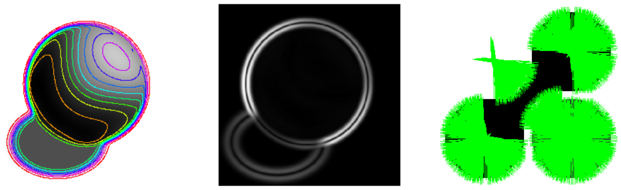
3. Isophote Centers

Since the curvature is the reciprocal of the radius, we can reverse Eq. (5) 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 (Figure 2(a)), 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

$$\begin{aligned} D(x,y) &= \frac{\{L_x, L_y\}}{L_w}\left(-\frac{L_w}{L_{vv}}\right) = -\frac{\{L_x, L_y\}}{L_{vv}} \\ &= -\frac{\{L_x, L_y\}(L_x^2 + L_y^2)}{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}. \end{aligned} \tag{6}$$

where $D(x, y)$ are the displacement vectors to the estimated position of the centers, which can be mapped into an accumulator, hereinafter *"centermap"*. The set of vectors pointing to the estimated centers are shown in Figure 2(b). Note that two smaller centers are generated by the concavities of the shape. Figure 2(c) represents the cumulative vote of the vectors for their center estimate (*i.e.* the accumulator). 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. The contribution of each vector can be weighted according to a relevance mechanism, discussed in the following section.

[1] Scale in this context represents the standard deviation of the Gaussian kernel or its derivatives with which we convolve the image. See [11, 18] for more details.



(a) (b) (c)

Figure 3. The isophotes of a sphere image (a), the curvedness value of the same image (b), the displacement vectors obtained by the isophotes with high curvedness in the toy image (c)

3.1. Center Voting

So far we have used an edge-based approach and a toy example to ease the explanations. In this specific case, we can assume that there are only three isophotes in the image: two describing the shape of the objects and one describing the background. By convolving the toy image with a Gaussian kernel, it can be observed that: (1) the number of isophotes increases around the edges; (2) besides some creations and annihilations, every new isophote can be considered as a scaled instance of the isophote that was describing the object. Based on these observations we can safely consider every new isophote as a new edge, which can be used to reinforce the voting of a center. 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.

Contrary to the used toy example, in real world environments we have no guarantee that the boundaries of an object are of the same intensity, *i.e.* that there is a sole isophote under the object's edges. In this case, allowing every isophote to vote for a center will produce meaningless results, since the shape of the isophotes differs from the shape of the object (Figure 3(a)). In order to cope with this drawback, 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 an operator that yields more information about the relation between the isophotes and the boundaries of an object.

The curvedness, indicating how curved a shape is, was introduced by [18] as:

$$curvedness = \sqrt{L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2}. \qquad (7)$$

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 (Figure 3(b)). 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. Figures 3(a) and 3(b) show the relation between the curvedness and the image isophotes. It is clear that the curvedness is higher where the isophotes are equally distributed. Figure 3(c) shows the displacement vectors obtained from every selected isophote in the toy image. Note the difference between Figures 2(b) and 3(c): in the former the centers are determined by a single isophote (sampled on the edges), while in the latter all relevant parts of the isophotes (selected through the curvedness) participate in the center voting. The advantage of the proposed 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.

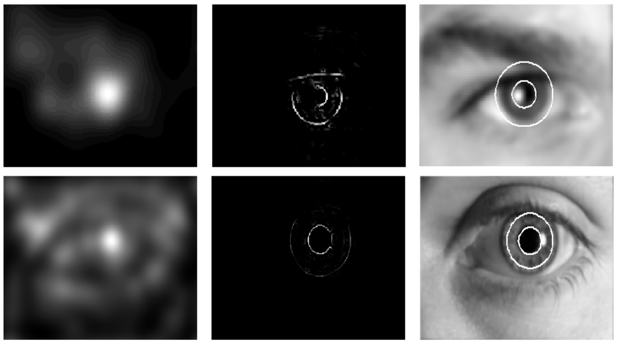


Figure 4. The obtained centermap, the edges that contributed to the vote, and an example of radius clustering

3.2. 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 (*i.e.* 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.

If one is interested in the most relevant radii (*i.e.* the cornea and the iris), the pixels that generated the MIC can be back projected to obtain a distribution of radius candidates, on which we can perform clustering. Figure 4 shows the results of the procedure applied on two high resolution eye images. Note from the back-projected edges that our method is able to cope with strong highlights (since bright votes are discarded) and blurred images (due to less aliasing). In general, however, we expect certain lighting conditions and occlusions from the eyelids to result in a wrong MIC. To cope with this problem, we propose two enhance-

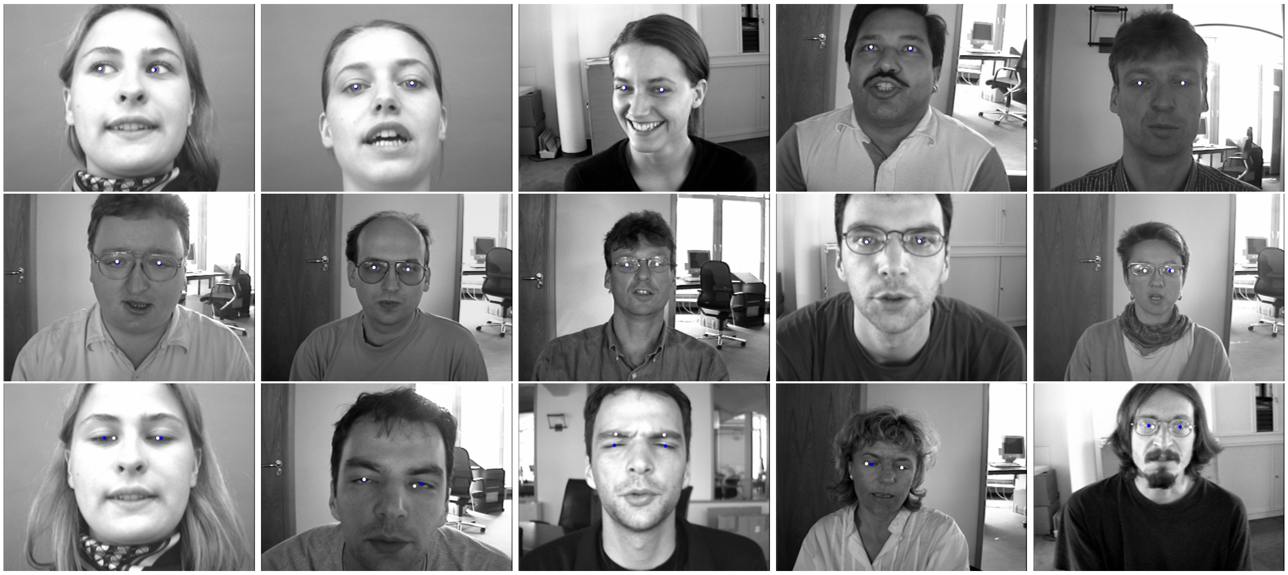


Figure 5. Sample of success and failures (last row) on the BioID face database; a white dot represents the estimated center.

ments to the basic approach, the first using mean shift for density estimation and the second using machine learning for classification.

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 (*e.g*. 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 distribution. After some iteration, the isocenter closest to the center of the search window is selected as the new eye center estimate.

Machine Learning: instead of focusing on a single MIC, the idea is to consider the n most relevant ones and to discriminate between them using a classification framework. Therefore, a window is cropped around each candidate isocenter (the size depends on the size of the detected face) and is scaled to a reference size. A SIFT [20] based descriptor (it differs from the SIFT as it does not search for scale invariant features, since we already know where the feature is) is computed at the center of the window (supposedly, the center of the eye). The resulting descriptor is then compared to a library of descriptors obtained from a training set, using a simple KNN classifier (experimentally this classifier achieved the best results). In the following section we will see how the proposed extensions compare to the basic approach.

4. Evaluation

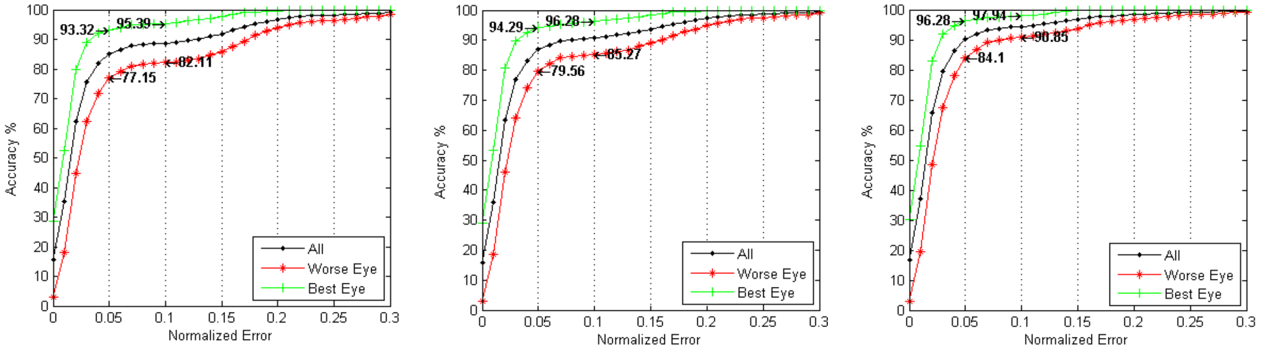
So far we used high resolution images of eyes as examples. In this section we test the proposed method on the more challenging task of locating eye centers on low resolutions face images, *e.g*. coming from a web cam. Additionally, we test the method for robustness in changes in pose and illumination, and for robust eye center tracking.

4.1. Eye Center Location: Accuracy

The used test set is the BioID database [4]. 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 (*i.e*. uncontrolled illumination). Besides changes in illumination, the position of the subjects changes both in scale and pose. Furthermore, in many samples of the dataset the subjects are wearing glasses. In some instances the eyes are closed, turned away from the camera, or completely hidden by strong highlights on the glasses. Due to these conditions, the BioID dataset is often considered a “difficult” dataset. The size of each image is 384x288 pixels. A ground truth of the left and right eye centers is provided with the dataset.

For each subject, the face position is estimated by using the boosted cascade face detector proposed by Viola and Jones [24]. The rough positions of the left and right eye regions are then estimated using anthropometric relations. The proposed procedure is then applied to the cropped eye regions (approximately 40x30 pixels) in order to accurately locate the center of the eye.

The *normalized error*, indicating the error obtained by the worse eye estimation, is used as the accuracy measure for the found eye locations. This measure was proposed by



(a) MIC

(b) MIC+MS

(c) MICs+ SIFT KNN

Figure 6. Accuracy vs. minimum (best eye) and maximum (worse eye) normalized error obtained with the proposed methods. "All" indicates the average.

Jesorsky *et al.* [16] and is defined as:

$$e = \frac{\max(d_{left}, d_{right})}{w}, \tag{8}$$

where d_{left} and d_{right} is the Euclidean distance between the located eyes and the ones in the ground truth, and w is the Euclidean distance between the eyes in the ground truth. In this measure, $e \leq 0.25$ (a quarter of the interocular distance) roughly corresponds to the distance between the eye center and the eye corners, $e \leq 0.1$ corresponds to the range of the iris, and $e \leq 0.05$ corresponds the range of the cornea. In order to give upper and lower bounds to the accuracy, in our graphs (Figure 6) we also show the *minimum normalized error*, obtained by considering the best eye estimation only, and an average between the best and worse estimation (indicated by "All").

Figure 5 shows the results obtained on different subjects of the BioID database. 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.

The graph in Figure 6(a) shows the accuracy of our method for different e. While it is clear that most of the results are nearly optimal, there is a saddle on the normalized error around the value of 0.15. This cluster of errors is consistent with the previous observations: small errors occur between the real eye centers and the eye corners/eyebrows. The improvement obtained by using the mean shift procedure for maximum density can be seen by comparing the graphs in Figures 6(a) and (b). Without any additional constraint, the results improved with $\approx 2.5\%$ over the basic approach, and the variance between the best and worse eye location is considerably reduced. The graph in Figure 6(c) shows the accuracy obtained by using the KNN classifier to discriminate between the top MICs, which achieved better results than both the basic and the mean shift approaches. However, note that by using classification the successful outcome of the system will inherently depend on the conditions it was trained with.

4.2. Illumination and Pose: Robustness

To systematically evaluate the robustness of the proposed eye locator to lighting and pose changes, two subsets of the Yale Face Database B [14] are used. The full database contains 5760 grayscale images of 10 subjects each seen under 576 viewing conditions (9 poses x 64 illuminations). The size of each image is 640x480 pixels. To independently evaluate the robustness to illumination and pose, we test the system on frontal faces under changing illumination (10 subjects x 64 illuminations) and on changing pose under ambient illumination (10 subjects x 9 poses).

Figure 7 shows a qualitative sample of the results obtained for one subject in the illumination subset. By analyzing the results, we note that the system is able to deal with light source directions varying from $\pm 35°$ azimuth and from $\pm 40°$ elevation with respect to the camera axis. The results obtained under these conditions are shown in Table 1. For higher angles, we note that the method is often successful for the less illuminated eye and sporadically for the most illuminated one: if the eye is uniformly illuminated, its center is correctly located, even for very low intensity images; if, on the other hand, the illumination influences only parts of the eye (some instances in the second row in Figure 7), the shape of the isophotes is influenced by shadows, resulting in an unreliable MIC.

Figure 8 shows the results of the eye locator applied to the pose subset of the Yale Face Database B. Also here we can see the robustness of our approach: due to the higher resolution and the absence of occlusions and glasses, every method obtained an accuracy of 100.00% for $e \leq 0.05$. The first errors are actually obtained considering $e \leq 0.03$, where the system achieves an accuracy of 93.18%.

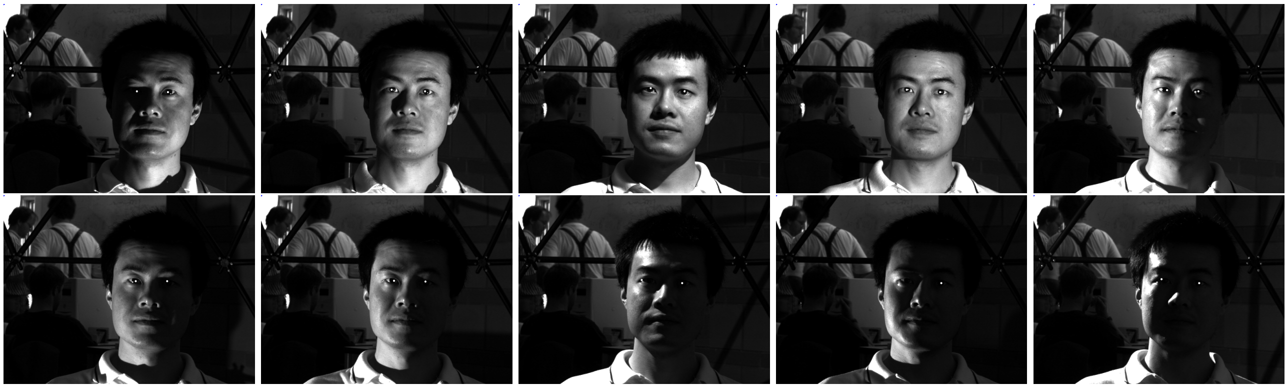


Figure 7. Results changing the illumination on a subject of the Yale Face Database B.

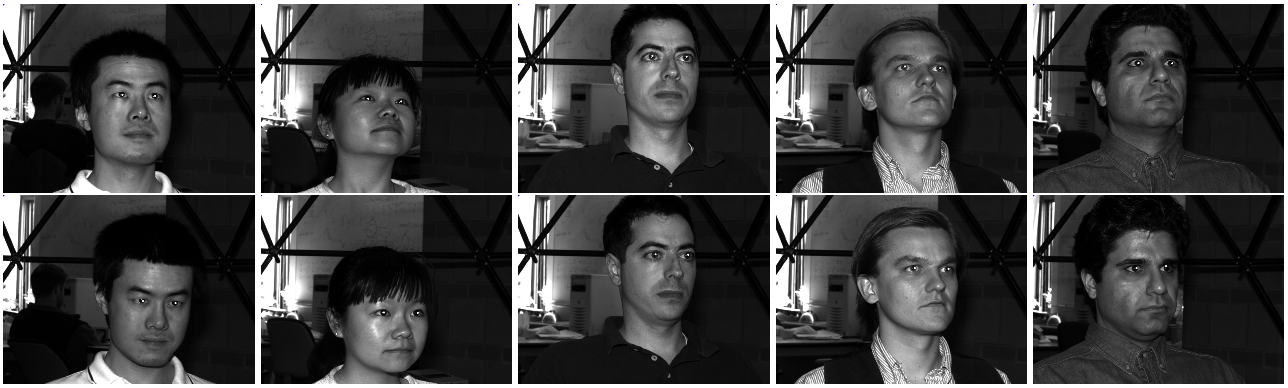


Figure 8. Sample of results on the Yale Face Database B (ambient illumination, changing pose).

Method	Accuracy ($e \leq 0.05$)	Accuracy ($e \leq 0.10$)	Accuracy ($e \leq 0.25$)
MIC	74.01%	82.57%	87.77%
MIC+MS	77.06%	85.63%	90.21%
MICs+SIFT	77.98%	86.85%	89.60%

Table 1. Accuracy vs. normalized error for illumination changes.

4.3. Comparison with the State of the Art

We compare our results with state of the art methods in the literature which use the BioID database and the same accuracy measure. The method used by Asteriadis *et al*. [1] assigns a vector to every pixel in the edge map of the eye area, which points to the closest edge pixel. The length and the slope information of these vectors is consequently used to detect and localize the eyes by matching them with a training set. Jesorsky *et al*. [16] use a face matching method based on the Hausdorff distance followed by a Multi-Layer Perceptron (MLP) eye finder. Cristinacce *et al*. [10, 9] use a multistage approach to detect facial features (among them the eye centers) using a face detector, Pairwise Reinforcement of Feature Responses (PRFR), and a final refinement by using Active Appearance Model (AAM) [8]. Türkan *et al*. [22] use edge projection (GPF) [25] and support vector machines (SVM) to classify estimates of eye centers. Bai *et al*. [2] use an enhanced version of Reisfeld's *generalized symmetry transform* [21]) for the task of eye location. Hamouz *et al*. [15] search for ten features using Gabor filters, use features triplets to generate face hypothesis, register them for affine transformations and verify the remaining configurations using two SVM classifiers. Finally, Campadelli *et al*. [5] use an eye detector to validate the presence of a face and to initialize an eye locator, which in turn refines the position of the eye using SVM on optimally selected Haar wavelet coefficients.

Table 2 shows the comparison between our methods and the methods mentioned above for an allowed normalized error smaller than 0.05, 0.1 and 0.25, respectively. Where inexplicitly reported by the authors, the results are estimated from their normalized error graphs, safely rounded up to the next unit. It can be seen that, for an allowed normalized error smaller than 0.25, we achieved accuracy comparable to the best methods. For iris location ($e \leq 0.1$), our method achieved superior accuracy with respect to the other methods, except for the one used by Cristinacce *et al*. [10]. This can be justified by the fact that the method used by Cristinacce uses other facial features to estimate and adjust the

Method	Accuracy	Accuracy	Accuracy
	$(e \leq 0.05)$	$(e \leq 0.10)$	$(e \leq 0.25)$
MIC	77.15%	82.11%	96.35%
MIC+MS	79.56%	85.27%	97.45%
MICs+SIFT	**84.10%**	**90.85%**	**98.49%**
Asteriadis [1]	**74.00%***	81.70%	97.40%
Jesorsky [16]	40.00%	79.00%	91.80%
Cristinacce [10]	56.00%*	**96.00%**	98.00%
Türkan [22]	19.00%*	73.68%	**99.46%**
Bai [2]	37.00%*	64.00%	96.00%
Campadelli [5]	62.00%	85.20%	96.10%
Hamouz [15]	59.00%	77.00%	93.00%

Table 2. Accuracy vs. normalized error for different methods. *=value estimated from author's graphs

position of the eyes (*i.e.* the eye center is in between the eye corners) which works extremely well for correct iris location, but does not have enough information to locate the exact eye center. However, our approach excels for accurate eye center location ($e \leq 0.05$), even when using the basic approach.

4.4. Eye Center Tracking

Even if the best accuracy is obtained by the MICs+SIFT method, applying it to video images thirty times per second will necessarily result in unstable estimates. However, the MIC+MS method scales perfectly to use temporal information: the converged position of the MS window can be used as initialization for the next frame, and the eye locator can be used to reinitialize the tracking procedure when it is found to be invalid (*i.e.* when the current MIC falls outside the mean shift window). This synergy between the two components allows the tracking system to be fully autonomous and user independent. The tracking performances are evaluated by using image sequences collected in an uncontrolled environment. The videos were obtained using a low-resolution web cam and ten different subjects were asked to test the system by performing fast movements, pose, and scale variation. For each sequence, we consider a track to be valid if it ends inside the iris region ($e \leq 0.1$). Without considering the frames in which a face was not detected due to motion blur or extreme side poses, the MIC+MS tracker achieved an overall accuracy of 97.33%, while the frame-based MIC and MICs+SIFT approaches scored a lower 85.33% and 91.15%, respectively. Figure 9 qualitatively shows the tracking results obtained using the MIC+MS tracker. The white square indicates the search window for the mean shift tracker, while the green line indicates the tracking path in the last 20 frames. To better understand the robustness of the tracker, note that in most of the screenshots the green track is a dot, while in the last image a fast head movement produces two R-shaped tracks. On the second row it is also possible to see the failure of the tracker when applied to a subject wearing glasses in the presence of strong highlights and the success on the same subject when the highlights do not appear on the glasses.

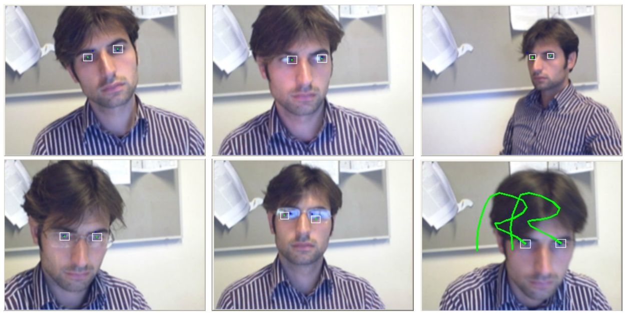


Figure 9. Sample of tracking results

4.5. Discussion

An additional advantage of the proposed approach is its low computational complexity: since the basic system (without classification) only requires the computation of image derivatives which is linear in the size of the image and the scale ($O(\sigma N)$), it allows for a real-time implementation. On a 2.4GHz Intel Core 2 Duo, using a single core implementation, the system was able to process ≈ 2500 eye regions per second on a 320x240 image. Including the face detector and the mean shift procedure, the algorithm takes 11ms per frame, which roughly corresponds to 90 frames per second, therefore the final frame rate is only limited by the web cam's frame rate. Due to the high accuracy and low computational requirements of the proposed method, we believe that it can be successfully used as a preprocessing step to some of the other systems. In particular, systems using classifiers (*e.g.* [5, 16, 22]) should benefit from the reduction in the search and learning phases and can focus on how to discriminate between few candidates. Furthermore, note that our system does not use any heuristics or prior knowledge to discriminate between candidates. We therefore suggest that it is possible to achieve superior accuracy by integrating the discussed method into systems using contextual information (*e.g.* [10, 15]).

However, even if the proposed method compares and updates the state of the art, it has certain limitations: the accuracy of the system is in fact conditioned by the presence of a symmetrical circular feature (*e.g.* the eye). If the eye is closed, occluded by highlight or wrongly illuminated, the algorithm will not be able to correctly locate an isocenter where the eye is supposed to be. To understand what the maximum accuracy achievable by our method is, we computed the normalized error obtained by selecting the isocenter closest to the ground truth. The graph in Figure 10 shows the comparison between the basic method and the proposed

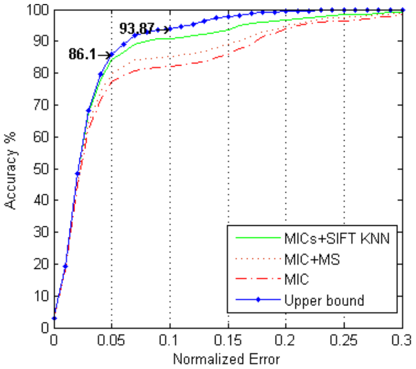


Figure 10. A summary of the obtained results, in comparison with the upper bound curve for our method

extensions, and an additional curve which represents the found upper bound on the BioID database. We can see that the extensions helped to increase the bending point of the curve, while the rest of the curve is similar in all the cases. This means that the extensions reduced the number of times an eye corner or an eyebrow is detected as the MIC, moving the results closer to the upper bound. Note that the SIFT extension almost follows the upper bound for $e \leq 0.05$.

5. Conclusions

In this paper, we proposed a new method to infer eye center location using circular symmetry based on isophote properties. For every pixel, the center of the osculating circle of the isophote is computed from smoothed derivatives of the image brightness, so that each pixel can provide a vote for its own center. The use of isophotes yields low computational cost (which allows for real-time processing) and robustness to rotation and linear illumination changes.

An extensive evaluation of the proposed approach was performed, testing it 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 our method is able to achieve highest accuracy, but this is somewhat bounded by the presence of the eye pattern in the image. We suggest the integration of our approach into existing contextual systems to combine their strengths.

References

[1] 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.

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

[3] 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.

[4] BioID Technology Research. The BioID Face Database. http://www.bioid.com, 2001.

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

[8] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.

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

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

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

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

[13] B. Froba, A. Ernst. Face detection with the modified census transform. *Aut. Face and Gest. Recog.*, pp. 91–96, 2004.

[14] A. Georghiades, P. Belhumeur, D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *PAMI*, 23(6):643–660, 2001.

[15] 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.

[16] 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.

[17] C. Kervrann, M. Hoebeke, and A. Trubuil. Isophotes selection and reaction-diffusion model for object boundaries estimation. *IJCV*, 50:63–94, 2002.

[18] J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Im. Vis. Comp.*, pages 557–565, 1992.

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

[20] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 20:91–110, 2003.

[21] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context free attentional operators: the generalized symmetry transform. *IJCV*, 14:119–130, 1995.

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

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

[24] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

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

[26] Z. Zhu and Q. Ji. Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *CVIU*, 98(1):124–154, 2005.