

Adapting *Starburst* for Elliptical Iris Segmentation

Wayne J. Ryan, Damon L. Woodard, Andrew T. Duchowski, and Stan T. Birchfield

Abstract— Fitting an ellipse to the iris boundaries accounts for the projective distortions present in off-axis images of the eye and provides the contour fitting necessary for the dimensionless mapping used in leading iris recognition algorithms. Previous iris segmentation efforts have either focused on fitting circles to pupillary and limbic boundaries or assigning labels to image pixels. This paper approaches the iris segmentation problem by adapting the *Starburst* algorithm to locate pupillary and limbic feature pixels used to fit a pair of ellipses. The approach is evaluated by comparing the fits to ground truth. Two metrics are used in the evaluation, the first based on the algebraic distance between ellipses, the second based on ellipse chamfer images. Results are compared to segmentations produced by ND_IRIS over randomly selected images from the Iris Challenge Evaluation database. Statistical evidence shows significant improvement of *Starburst*'s elliptical fits over the circular fits on which ND_IRIS relies.

I. INTRODUCTION

Except for several relatively unique approaches, e.g., [3], [16], common iris segmentation methods model the iris as a pair of circles [5]. Although the inner and outer boundaries of the iris may be roughly approximated by circles, they rarely appear as true circles in images [9]. The iris image is subject to perspective projection. It is approximately planar. Any circle that lies in a plane not fronto-parallel to the camera will appear elliptical in the image plane. The segmentation model must account for such distortions. A general ellipse model is therefore more appropriate than a restricted circular model to compensate for this type of distortion.

The *Starburst* algorithm was introduced by Li, Babcock, and Parkhurst for the purpose of eye tracking [14]. For such an application, *Starburst*'s main objective is to identify feature points on the limbus for subsequent localization of the pupil center. *Starburst* then fits an ellipse to the limbic pixels, operating under the implicit assumption that the center of that ellipse coincides with the pupil center. The pupil center is then used for estimating the point of gaze, or POG, of a viewer wearing the eye tracking apparatus.

In this paper we adapt the *Starburst* algorithm for the purpose of iris segmentation. The novelty behind our adaptation is the simultaneous identification of both pupillary and limbic boundaries, fitting ellipses to both contours, thereby producing an iris segmentation suitable for subsequent iris

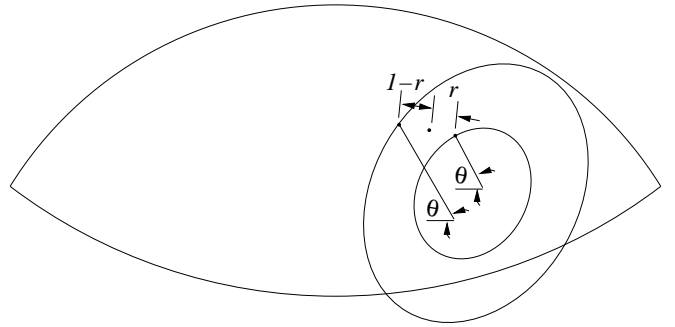


Fig. 1. Modeling iris segmentation (patterned on actual image, see Fig. 13).

recognition as well as eye tracking applications. Such contour fitting is an essential component of iris recognition [8].

We also present a technique for eyelid detection. It has been shown that localization of the eyelid improves accuracy, reliability, and efficiency by reducing the search area for both pupillary and limbic features and by eliminating distracting features like eyelashes [22]. We utilize active contours to detect the eyelids and demonstrate improved accuracy of iris segmentation.

The third contribution of this paper is the introduction of two comparisons to ground truth for evaluating the contour fitting algorithm: the first based on the root sum squared metric of algebraic distance between fit ellipses, the second based on comparison of ellipse chamfer images. We use these metrics to compare our algorithm's elliptical segmentations to those of ND_IRIS, a readily available iris segmentation algorithm [17]. We test images randomly selected from the publicly available Iris Challenge Evaluation (ICE) database. We present strong statistical evidence showing improved elliptical fit accuracy of our approach over ND_IRIS. Computation time requirements of the two algorithms are roughly equivalent.

II. BACKGROUND

Daugman [7] models the iris as an elastic sheet stretched between the pupil and limbus contours, assigning a pair of dimensionless coordinates (r, θ) to each pixel at (x, y) , as shown in Fig. 1. This mapping can be represented as $I(x, y) \rightarrow I(r, \theta)$ where $x(r, \theta)$ and $y(r, \theta)$ are linear combinations between the pupillary boundary and the limbus, $I(r, \theta) = (1 - r)I_p(\theta) + rI_s(\theta)$, with I_p and I_s denoting pixels along the pupil and limbus contours, respectively.

Image segmentation algorithms may be classified as either labeling or fitting. We consider labeling algorithms to be those that segment an image into groups by assigning labels indicating to which group a pixel belongs. Examples include

W. J. Ryan, D. L. Woodard, and A. T. Duchowski are with the School of Computing, Clemson University, Clemson, SC 29634, USA {wryan|woodard|aduchow}@clemson.edu

S. T. Birchfield is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA stb@clemson.edu

graph cuts, *level sets*, and *watershed*. We consider fitting algorithms to be those that fit a parametrized model to image pixels. Examples include the *Hough transform*, *snakes*, and *Starburst*.

Daugman’s elastic sheet model necessitates the use of a curve fitting algorithm [8], e.g., snakes [9]. ND_IRIS uses the Hough transform for segmentation [17]. We present the applicability of Starburst.

In [13] the ideas of ray casting, locating two feature points per ray, and filtering by distance were introduced. In [20] Starburst was augmented with luminance delineation. In this paper we combine these notions into a single algorithm that is capable of accurately fitting ellipses to both the pupillary and limbic boundaries.

A simple ray-based algorithm resembling Starburst was used as a post processing step in a graph cuts approach to iris segmentation, presumably to bolster results when graph cuts failed [19]. As a labeling algorithm, graph cuts does not support the elastic sheet model. Its utility is limited to the creation of a mask as is currently done by statistical inference [9]. No comparison between graph cuts and statistical inference was made in [19], however. Sufficient detail of the ellipse fitting algorithm was also lacking.

III. ELLIPTICAL IRIS SEGMENTATION

Starburst is a randomized local search algorithm used in eye tracking. Starburst was developed by Li *et al.* [15] to compensate for the high degree of noise present in low cost off-the-shelf cameras. The original algorithm proved to be a stable way to track the eye under NIR illumination yet there were three main sources of error: the algorithm had difficulty distinguishing between the pupillary boundary and the limbus, specular reflections caused erroneous feature points, and eyelashes and eyelids introduced noise and occlusion.

By incorporating luminance information Starburst was better able to distinguish between the pupillary and limbic boundaries [20]. Independently in [13], the start point of the rays was constrained to any location on the pupil and each ray generated two feature points. The point closest to the origin of the ray would be assigned rank of one and the farther point a rank of two. This rank assignment helps distinguish between pupillary and limbic boundaries.

Luminance and rank based delineation of feature points may be combined to discard many feature points that result from specular reflections. In this paper we implement a snakes algorithm to find upper and lower eyelid boundaries. These boundaries mask out the feature search neighborhood beyond the eyelids and eyelashes.

A. General Description

Our adaptation of Starburst, outlined in Fig. 2, requires an image of an eye and coordinates of an initial point near the pupil center. Rays are cast away from the initial point in a star-like pattern. The gradient is calculated along each ray and used to identify feature points on the pupillary and limbic boundaries. These feature points are used to randomly compute a (potentially large) number of ellipses.

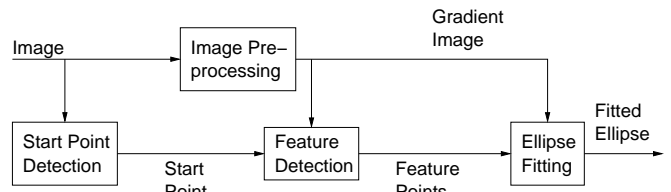


Fig. 2. Algorithm flow.



Fig. 3. Eye image after thresholding (left) and chamfer operations (right).

These ellipses are evaluated and the best are selected as the pupillary and limbic contours.

B. Detailed Description

1) *Image Pre-processing*: We preprocess the image by convolution with Gaussian filters. We first use a simple smoothing filter then a gradient detection filter. We use the resulting gradient vectors for both feature detection and ellipse fitting. Starburst requires an initial location from which to begin searching. The ICE database does not provide such initial points. We therefore begin with a simple threshold algorithm that locates a seed point.

2) *Start Point Detection*: We find the start point in two steps. First, the darkest 5% of pixels are set to black; all others are set to white. This is done to isolate the pupil as it is a dark part of the image and covers slightly less than 5% of the image (see Fig. 3, left). We find that eyelashes and other hair are often as dark as the pupil, but cover very little area.

Second, we calculate the chamfer image using 3, 4 weighting [4] so that the darkest pixel is the pixel farthest from any white pixel (see Fig. 3, right). Since eyelashes and other hair are long and thin while the pupil is round the location of the darkest pixel is most likely within the pupillary boundary and a good start point for the remainder of our algorithm.

This simple thresholding algorithm will fail when the eyelashes are accentuated by heavy mascara and the pupil is simultaneously occluded by bright specular reflections. Nevertheless it is effective on a majority of the ICE database.

3) *Feature Detection*: We use dot products to calculate the component of the gradient collinear with rays pointing radially away from the start point (see Fig. 4, left). The objective is to mark feature points at the locations where the rays exit dark regions. Each ray marks feature points at the two largest gradient peaks within an experimentally determined epsilon distance. The term rank is used to indicate which of the two points is closer to the origin of the ray. A rank 1

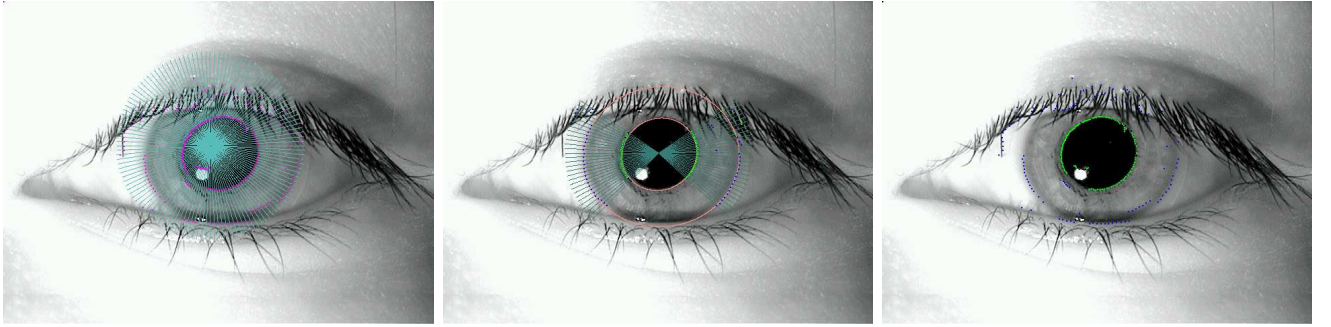


Fig. 4. Rays used to detect feature points (left), with point culling due to hardcoded *a priori* constraint on ray direction when eyelid detection is not used (middle), and classified feature points (right; with pupil green, limbus blue, junk black).

feature point is closer to the origin and expected to be on the pupillary boundary. A rank 2 feature point is further from the origin and expected to be on the limbus.

Feature detection without lid detection (see below) assumes *a priori* that eyelids exist above and below the start point and that eyelids occlude the top and bottom portions of the limbus. This assumption leads to (hardcoded) culling of the top 1/3 and bottom 1/4 of candidate feature points, following sorting by their y -values (see Fig. 4, middle). As a result, these top and bottom feature points are not used for subsequent ellipse fitting.

Surviving feature points are classified into three categories: pupil, limbus, and junk. We begin by sorting the points by their corresponding luminance. We expect 50% of them to be on the pupil. The luminance of pixels on the pupil is less than those on the limbus. Any feature point of rank 1 with low luminance is labeled as a pupil point. Those that are of rank 2 with high luminance are labeled as limbus points. All others are labeled as junk (see Fig. 4, right).

4) *Ellipse Fitting*: Once we have detected and classified our feature points we fit ellipses to the feature set. By selecting five pupil points at random we can create a 5×5 system of equations from the general quadratic expression for an ellipse:

$$ax^2 + by^2 + cx + dy + exy + f = 0. \quad (1)$$

We set the f coefficient to an arbitrary value and use Gaussian elimination to compute the remaining coefficients.

Once an ellipse is generated it must be evaluated. We generate many such ellipses, evaluate them all, and take the mean value of the best few to be our final contour. It may seem strange to average several rather than simply retaining the single best. Yet the average yields better results due to the randomized nature of the algorithm. There is some degree of independence between the results and each is affected by random noise in the input. The resulting random error may be minimized by averaging the result of multiple trials.

The evaluation of the ellipse is equal to the mean evaluation of all the pixels through which it passes. A good pixel is one that is on or near the peak of a strong gradient pointing toward the center of the ellipse. First, we detect edges in the image with the Canny edge detector to find pixels on a peak

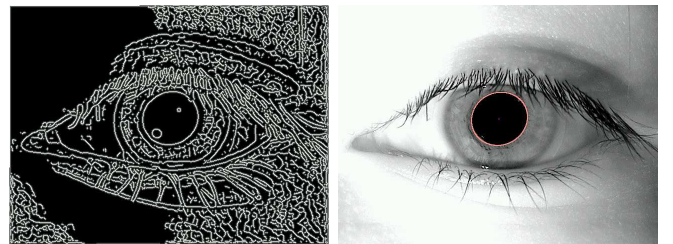


Fig. 5. Edge image (left) and ellipse fit to pupil (right).

gradient. Next, we blur the edge detected image slightly to find pixels near a peak. For each pixel we compute the dot product of the unit vector pointing from that pixel toward the center of the ellipse and the gradient at that pixel. This is multiplied by the corresponding pixel in our edge image (see Fig. 5).

The following expression describes our ellipse evaluation where $\nabla(x, y)$ is the gradient, $E(x, y)$ is the edge value, the unit vector $\nu(x, y)$ points toward the ellipse center at pixel location (x, y) , and the ellipse passes through n pixels:

$$\sum_{\forall(x,y) \text{ on ellipse}} \frac{E(x, y) (\nabla(x, y) \cdot \nu(x, y))}{n}$$

Note that the solution of the 5×5 system of equations and the subsequent evaluation of the generated ellipse may be computationally prohibitive if the number of systems solved and ellipses evaluated is very high. We improve the probability of generating a good ellipse by selecting feature points in an intelligent way.

We have noticed that inferior combinations of points include points that are spatially clustered. We have implemented a heuristic algorithm that encourages selection of points that are spatially distant (see pseudo-code in Alg. 1).

C. Eyelid Detection Algorithm

We use the snake algorithm to detect eyelids. Snakes, first introduced by Kass *et al.* [12], can be used to extract contours from images, or track objects in video. The location of the contour is determined by an ordered set of control points. The snake algorithm minimizes an energy function that depends

```

FeatureSelect(feature point set  $S$ )
  limit = 10
  while  $|S| < 5$ 
    select a point  $P'$  at random
    for all points  $P \in S$ 
      if  $\angle P' - \angle P < \pi/\text{limit}$ 
        discard  $P'$  and pick a new one
        limit += 0.5
      else add  $P'$  to set  $S$ 

```

Alg. 1. Algorithm for selecting feature points, with \angle denoting the angle of the ray that detects point P .

on the snake's position:

$$energy = \sum_{\text{all edges}} data + elasticity + stiffness.$$

We consider *data* to be large where the image gradient magnitude is small. For each control point we define an initial and terminal location. The *elasticity* term of our function is large when the control points are far from their respective terminal positions. The final term enforces our assumptions about the shape of the contour. We define *stiffness* to be large when the curvature deviates from what we would expect of an eyelid. We allow control points to move only in the vertical direction.

The algorithm is implemented through the use of dynamic programming [1]. A $9 \times n$ table of $(prev, energy)$ pairs is populated. Each column in the table corresponds to a pair of control points (v_i, v_{i+1}) , and each row corresponds to possible movement $((a, b), (c, d))$ for the points. A particular entry in column i row j contains the best total *energy* for all points 1 through i if points v_i, v_{i+1} move as indicated by row j . Note that only three rows of the $i - 1$ column are consistent with any particular entry of column i . This is because both columns contain v_i . These three consistent rows correspond to the three possible movements of v_{i-1} .

We compute the energy of entry i, j as

$$energy_{i,j} = \|\nabla(v_i^{\text{at}(a,b)})\| + \alpha \|v_i^{\text{at}(c,d)} - fn(v_i)\|^2 + \min_{k:k \diamond j} \left(\beta \|v_{i+1}^{\text{at}(c,d)} - 2v_{i-1}^{\text{at}(a,b)} + v_{i-1}^{\text{at}k}\|^2 + \Theta(k) \right),$$

where α and β are experimentally determined constants ($\alpha = \beta = 0.25$ in the current implementation), $\Theta(k)$ is the *energy* entry from consistent rows of the previous column, $k \diamond j$ means k is consistent with j , $fn(v_i)$ is a terminal location of v_i , and $\|\nabla(v_i^{\text{at}(a,b)})\|$ is the gradient magnitude. The first term is the *data* term, the second is the *elasticity*, and the third is the *stiffness*. An example of the resultant eyelid detection is shown in Fig. 6.

D. Combined Algorithm

Now that we have described all the major components of our iris segmentation algorithm we explain how they are assembled into a complete iris segmentation system. The refined algorithm flow is shown in Fig. 7.

The first step of the algorithm is to find a start point on the pupil. This is done using luminance threshold combined with

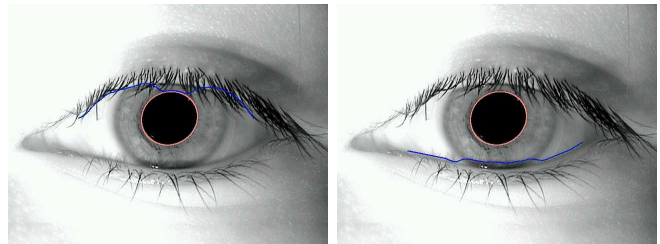


Fig. 6. Detection of upper (left) and lower eyelid (right).

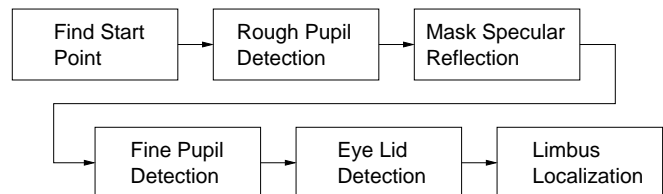


Fig. 7. Combined algorithm flow.

chamfer distance. Although this start point is nearly always on the pupil, specular reflections usually push it away from center.

As the pupil is more salient in the image than the limbus we locate its boundary first. The off-center nature of our start point introduces a bias in our pupil localization. This first invocation of the Starburst algorithm is sufficient for two purposes. First, the center of the resulting ellipse is much closer to the pupil center than our initial start point. Second, we are able to scale our ellipse down slightly and mask out the specular reflection. Scaling is easily accomplished by slightly increasing the constant term of our coefficients.

Masking is accomplished by setting values of all gradient vectors located within the ellipse to zero. We then run a second iteration of Starburst with the new start point, and with the masked image as input. This second iteration produces a better fit because the feature points are more evenly distributed around the appropriate boundary rather than clustering mostly to one side.

Next we use snakes to find the eyelids. The snakes do not precisely locate the eyelids, instead they are used to mask out areas where we are unable to find good limbic feature points. After the eyelids are detected we run a final iteration of Starburst to locate the limbus. Feature points beyond the eyelids are discarded and only pixels between the eyelids are used during ellipse evaluation. An example of the final segmentation is shown in Fig. 8.

It should be noted that our implementation of snakes often fails to properly mask the eyelids and eyelashes and that in most cases hardcoded constraints relating the eyelid position to the pupil location work just as well. Nevertheless, the fact that the automatic snake algorithm performs as well as our hardcoded *a priori* constraint on ray direction (as seen in the experimental results below) demonstrates the potential of this approach.

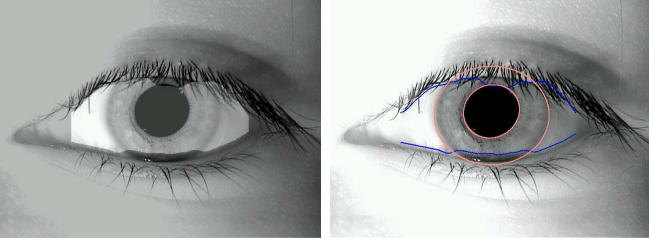


Fig. 8. Masked feature search region resulting from automatic eyelid detection (left) and final segmentation (right).

IV. EVALUATION OF THE ALGORITHM

We have manually segmented 245 images from the ICE database with closed contours modeled by ellipses, to serve as ground truth for comparison of Starburst to ND_IRIS. Low contrast between pupil and iris might pose a potential problem. Images poorly suited to either approach were not excluded from the random sample, e.g., see Fig. 11.

Although other experimental efforts report overall biometric accuracy measures such as equal-error or rank-one recognition rates [5], some of which are subjective in nature, e.g., [18], here we concentrate specifically on objective elliptical goodness of fit of automatically segmented ellipses to ground truth. To do so, we introduce two distance metrics. The first is based on a closed form evaluation of the algebraic distance, the second is based on chamfer image segmentation of both fitted and ground truth images.

A. Evaluation Metric Based on Algebraic Distance

The quadratic equation (1) represents a generic conic as the zero set of a second order polynomial:

$$H(\mathbf{a}; \mathbf{x}) = \mathbf{a} \cdot \mathbf{x} = ax^2 + by^2 + cx + dy + exy + f = 0,$$

with $\mathbf{a} = [a \ b \ c \ d \ e \ f]^T$ and $\mathbf{x} = [x^2 \ y^2 \ x \ y \ xy \ 1]^T$. $H(\mathbf{a}; \mathbf{x}_i) = D$ is called the *algebraic distance* of a point \mathbf{x}_i to the conic $H(\mathbf{a}; \mathbf{x}) = 0$ [10]. Our ellipse distance metric is defined as the root sum squared (RSS) of algebraic distances of the sampled points of the tested ellipse w.r.t. the reference ellipse, i.e.,

$$\sqrt{\int_{x_{\min}}^{x_{\max}} H(\mathbf{a}; \mathbf{x}_i)^2 dx} \approx \sqrt{\sum_{x_{\min}}^{x_{\max}} H(\mathbf{a}; \mathbf{x}_i)^2}.$$

In practice, $\sum H(\mathbf{a}; \mathbf{x}_i)^2$ can be evaluated either directly, e.g., iterating over $x \in [x_{\min}, x_{\max}]$ for some small Δx , or following an approach similar to that of Bresenham [6]—an efficient, discretized scanline sampling of ellipse points usually employed for rendering. Our implementation evaluates $H(\mathbf{a}; \mathbf{x}_i)$ in a manner similar to Bresenham’s ensuring no gaps between sampled points on the tested ellipse.

This algebraic distance is a more robust metric of elliptical goodness of fit than simple average error of ellipse center and radii. Simple (Euclidean) distance measures of center displacement merely indicate translation error whereas the difference in radii reflects the elliptical orientation error. Reporting these separately (e.g., as in [19])

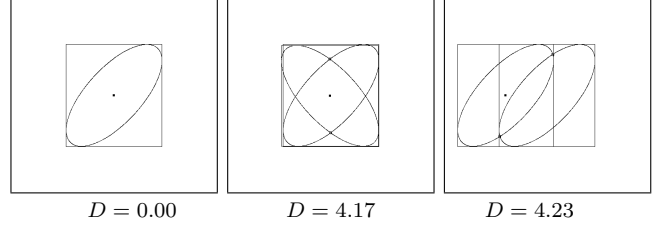


Fig. 9. Ellipse distance metric examples. In each case, the reference ellipse is situated at the origin with $r = 0.3$ and $s = 0.7$ rotated by $\theta = 40^\circ$ while a test ellipse (center indicated by a black dot) is rotated and/or shifted. Bounding boxes are drawn around each ellipse.

does not fully describe the misalignment between ellipses due to composite homographic transformation (including rotation, translation, and scaling of the ellipse axes). The above error metric takes into account the homographic transformation by virtue of evaluation of the second order quadratic, as the quadratic coefficients embody the ellipse’s rotation by θ about its center (h, k) , satisfying $(M(x-h) + N(y-k))^2 + (M(y-k) - N(x-h))^2 = r^2s^2$:

$$\begin{aligned} a &= s^2M^2 + r^2N^2 & c &= -2ha - ke \\ b &= s^2N^2 + r^2M^2 & d &= -2kb - he \\ e &= 2MN(s^2 - r^2) \\ f &= M^2(s^2h^2 + r^2k^2) + N^2(r^2h^2 + s^2k^2) + \\ & \quad 2MNHk(s^2 - r^2) - r^2s^2 \end{aligned}$$

where (r, s) are the lengths of the ellipse axes and $M = \cos \theta$ and $N = \sin \theta$. The RSS metric evaluates to 0 for exactly overlapping ellipses while producing non-zero values for rotated and translated ellipses, as shown in examples in Fig. 9.

B. Evaluation Metric Based on Chamfer Images

An analog of the above metric performed in image space can be obtained by creating a chamfer image for each segmentation such that each pixel indicates the distance from ground truth at that location. To evaluate the goodness of fit of a particular segmentation contour we introduce the *MDGT* (Mean Distance from Ground Truth). Let MDGT be defined as the mean value in the chamfer image of all pixels through which the contour passes.

Note that MDGT is somewhat similar to mean RSS error but operates in image space. Mean RSS is evaluated in ellipse coordinates rather than in image coordinates. We should note that a normalized form of RSS is also available, known as the Sampson error, which is a form of algebraic distance subject to Mahalanobis normalization [11].

C. Results

We have automatically segmented 245 images using each of three segmentation algorithms: Starburst without eyelid detection, Starburst with eyelid detection, and ND_IRIS. RSS and MDGT were computed for each automatically segmented image.

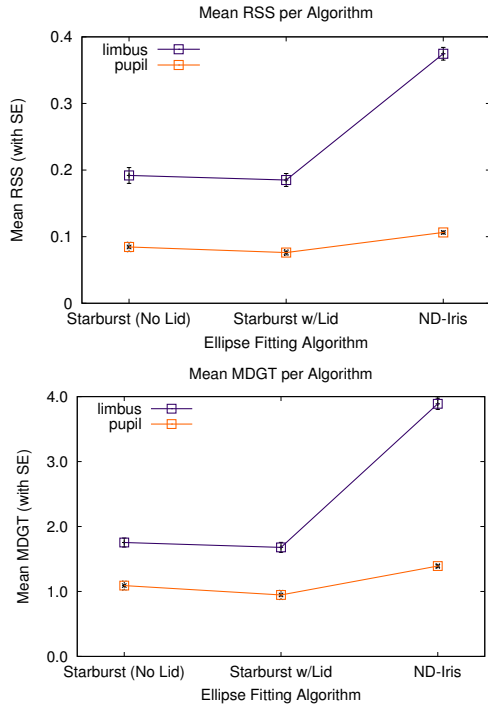


Fig. 10. Comparison of mean RSS (top) and MDGT (bottom) metrics.

Viewing the experiment as a 2×3 factorial design (2 fitted image features: pupil or limbus, and 3 algorithms: Starburst with and without lid detection and ND_IRIS) and considering the fitted image features and algorithms as fixed factors (with images as the random factor [2]), repeated-measures two-way analysis of variance, or ANOVA, indicates a significant main effect of feature on RSS ($F(1,244) = 555.13, p < 0.01$)¹ as well as a significant main effect of algorithm ($F(2,488) = 117.79, p < 0.01$), with feature \times algorithm interaction significant ($F(2,488) = 112.81, p < 0.01$).

Averaging across the three algorithms, pair-wise t-tests with pooled SD indicate significantly better performance of Starburst (with or without lid detection) over ND_IRIS ($p < 0.01$, with Bonferroni correction). Pair-wise t-tests show no significant difference between the two variants of Starburst.

Plotting the mean RSS and MDGT with standard error against algorithm type, as shown in Fig. 10, indicates that all three algorithms provide a statistically significant overall better fit to the pupil than to the limbus. Although use of eyelid detection shows no statistically significant advantage in its use by Starburst, on average, Starburst significantly outperforms ND_IRIS in both pupil and limbus ellipse fitting.

For readers unfamiliar with ANOVA, its tests are based on the F-ratio: the variation due to an experimental effect divided by the variation due to experimental error [21]. The null hypothesis assumes $F = 1.0$, or that the effect is the same as the experimental error, hence no significant difference is expected (between means of the sampled responses, assumed to be normally distributed). This hypothesis is rejected if the

¹Assuming sphericity as computed by R, the statistical analysis package used throughout.

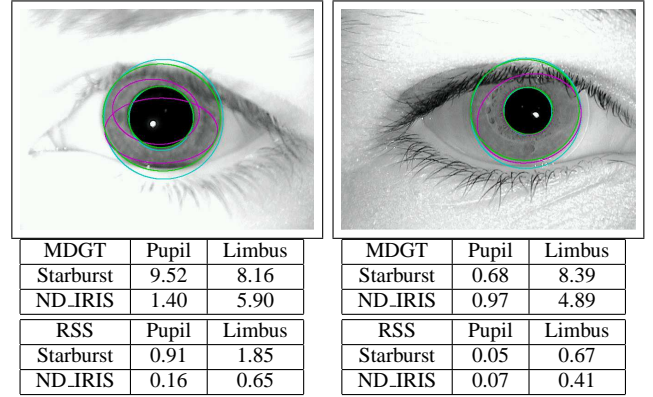


Fig. 11. Two worst Starburst segmentations.

F-ratio is significantly large enough that the possibility of it equaling 1.0 is smaller than some pre-assigned probability, e.g., $p = 0.01$, or one chance in 100, meaning that if $p < 0.01$ then the observed difference is $> 99\%$ certain to be solely due to experimental effect (the means are sufficiently far apart that the distributions do not overlap).

A critique of ANOVA for significance testing is the assumption of normality of the parametric data under inspection. The Kruskal-Wallis rank sum test is a nonparametric test that can be used in place of one-way ANOVA if the distribution is not normal. It is used in a similar manner as the Wilcoxon signed-rank test in place of the t-test. It is a test on the ranks of the original data and so the normality assumption is not required. Averaging across algorithms, the Kruskal-Wallis rank sum test indicates a significant difference in mean RSS ($\chi^2 = 195.36, df = 2, p < 0.01$). Similarly, averaging across limbus/pupil features, the Kruskal-Wallis rank sum test indicates a significant difference in mean RSS ($\chi^2 = 792.97, df = 1, p < 0.01$). The agreement between test significances simply shows that the normality assumption of ANOVA as used above is not unreasonable.

Similar significance results were obtained following ANOVA of the MDGT metric, as suggested in Fig. 10, but are omitted due to lack of space. Results from the three segmentations are visualized by displaying ground truth in green. Starburst with eyelid detection is displayed in magenta. ND_IRIS segmentation is displayed in cyan. Starburst without eyelid detection is similar to that with eyelid detection and is not shown. Image examples were selected by sorting the evaluation results by MDGT and selecting the best and worst few segmentations. From these images it is clear that high RSS and MDGT values correspond to poorer fits and, likewise, low RSS and MDGT values correspond to better fits.

The images in Fig. 11 illustrate the two worst Starburst fits. Note that the complete failure in the first image may be easily detected and compensated for by post processing. No part of the pupil ellipse should ever protrude outside the limbus ellipse.

The two best Starburst fits are shown in Fig. 12. Note that the magenta ellipse is partially occluded by the ground truth

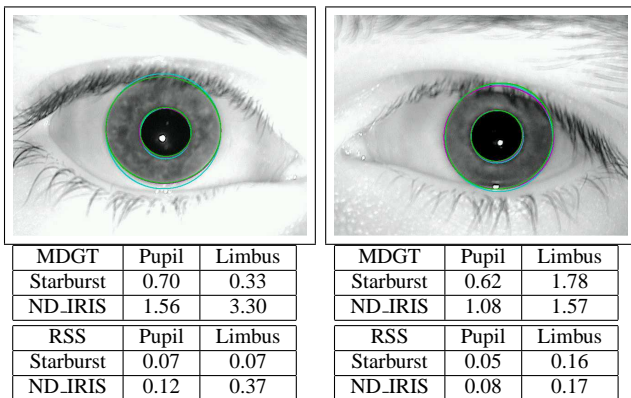


Fig. 12. Two best Starburst segmentations.

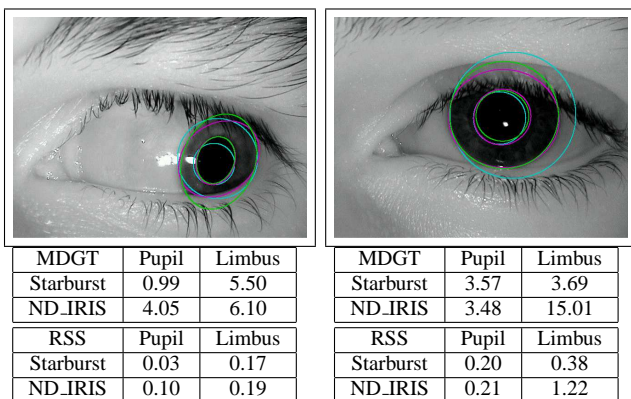


Fig. 13. Two typical Starburst segmentations.

ellipse. Notice also that the ND_IRIS segmentation deviates more at the top and bottom of the contour than at the sides. This is typical of ND_IRIS as it assumes circular models of the contours. Fig. 13 contains two more typical examples.

D. Discussion

The occasional failure of our implementation to properly segment the pupil can be attributed to failure of the simple thresholding algorithm to identify a good seed point and handle interference from specular reflections. The Hough transform used by ND_IRIS employs a global search and does not suffer from this problem. On average, however, adapted Starburst's elliptical fitting accuracy is superior to that of ND_IRIS. This is not surprising as the cause is likely due to Starburst's use of an elliptical contour model instead of a circular one. Since our metrics are a measure of elliptical goodness of fit, the circle is at a disadvantage. Thus, our analysis supports our hypothesis of the ellipse as more fitting for iris contour modeling than the circle.

Recall that our implementation without eyelid detection imposed *a priori* constraints on ray direction. Feature points in near vertical directions were not used for ellipse fitting. Consequently, the upper and lower portions of the limbus contour were omitted from fit evaluation. Automatic eyelid detection does not provide a significant accuracy benefit over hardcoded constraints, suggesting that both approaches are equally effective. Further refinement of the eyelid detection

algorithm should improve limbus segmentation.

V. CONCLUSION

A novel approach to iris segmentation based on the Starburst algorithm was given, showing significant improvement over ND_IRIS in fitting of the iris contours. Two metrics were introduced, one based on the algebraic distance between ellipses, the other on chamfer images. The ability of automatic eyelid detection via active contours to achieve results similar to hardcoded results suggests the potential of snakes for effective masking of feature points used for elliptical fitting. The resultant iris segmentation is thus suitable for subsequent iris recognition as well as eye tracking applications.

REFERENCES

- [1] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, September 1990.
- [2] J. Baron and Y. Li. Notes on the use of R for psychology experiments and questionnaires. Online Notes, 09 November 2007. URL: <<http://www.psych.upenn.edu/~baron/rpsych/rpsych.html>> (last accessed December 2007).
- [3] B. Bonney, R. Ives, D. Eitter, and Y. Du. Iris pattern extraction using bit planes and standard deviations. In *Thirty-Eighth Asilomar Conference on Signals, Systems, and Computers*, pages 582–586, November 2004.
- [4] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, 1986.
- [5] K. W. Bowyer, K. Holingsworth, and P. J. Flynn. Image understanding for iris biometrics: A survey. *Computer Vision and Image Understanding*, 110(2):281–307, 2008.
- [6] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.
- [7] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, November 1993.
- [8] J. Daugman. How Iris Recognition Works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, January 2004.
- [9] J. Daugman. New Methods in Iris Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(5):1167–1175, October 2007.
- [10] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct Least Squares Fitting of Ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.
- [11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2nd edition, 2003.
- [12] M. Kass, A. Witkins, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [13] D. Li. Low-Cost Eye-Tracking for Human Computer Interaction. Master's thesis, Iowa State University, Ames, IA, 2006. Techreport TAMU-88-010.
- [14] D. Li, J. Babcock, and D. J. Parkhurst. openEyes: A Low-Cost Head-Mounted Eye-Tracking Solution. In *ETRA '06: Proceedings of the 2006 Symposium on Eye Tracking Research & Applications*, San Diego, CA, 2006. ACM.
- [15] D. Li, D. Winfield, and D. J. Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *Vision for Human-Computer Interaction Workshop (in conjunction with CVPR)*, 2005.
- [16] X. Li. Modeling intra-class variation for non-ideal iris recognition. In *Springer LNCS 3832: International Conference on Biometrics*, pages 419–427, January 2006.
- [17] X. Liu, K. Bowyer, and P. Flynn. Experiments with an improved iris segmentation algorithm. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 118–123, 17-18 October 2005.
- [18] H. Prounça and L. A. Alexandre. Iris segmentation methodology for non-cooperative recognition. In *IEE Proceedings on Vision, Image and Signal Processing*, volume 153, pages 199–205, April 2006.

- [19] S. J. Pundlik, D. L. Woodard, and S. T. Birchfield. Non-Ideal Iris Segmentation Using Graph Cuts. In *Workshop on Biometrics (in conjunction with CVPR)*, 2008.
- [20] W. J. Ryan, A. T. Duchowski, and S. T. Birchfield. Limbus/pupil switching for wearable eye tracking under variable lighting conditions. In *ETRA '08: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, pages 61–64, New York, NY, 2008. ACM.
- [21] J. Tangren. A Field Guide to Experimental Designs. Online Notes, 22 April 2002. URL: <<http://www.tfrec.wsu.edu/ANOVA/index.html>> (last accessed May 2008).
- [22] G. Xu, Z. Zhang, and Y. Ma. Improving the performance of iris recognition system using eyelids and eyelashes detection and iris image enhancement. In *IEEE International Conference on Cognitive Informatics (ICCI)*, pages 871–876, 17-19 July 2006.