BioSystems 95 (2009) 130–136

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Automatic recognition and measurement of butterfly eyespot patterns

Margarida Silveira[a,b,*], Antónia Monteiro[c]

[a] Department of Electrical and Computer Engineering, IST University, Av. Rovisco Pais, 1049-001 Lisbon, Portugal
[b] Institute for Systems and Robotics, Lisbon, Portugal
[c] Department of Ecology and Evolutionary Biology, Yale University, New Haven, CT 06511, USA

ARTICLE INFO

Article history:
Received 9 March 2008
Received in revised form 2 September 2008
Accepted 12 September 2008

Keywords:
Computer learning
Butterfly eyespots
Automatic measurement
Eyespot recognition
Bicyclus anynana

ABSTRACT

A favorite wing pattern element in butterflies that has been the focus of intense study in evolutionary and developmental biology, as well as in behavioral ecology, is the eyespot. Because the pace of research on these bull's eye patterns is accelerating we sought to develop a tool to automatically detect and measure butterfly eyespot patterns in digital images of the wings. We used a machine learning algorithm with features based on circularity and symmetry to detect eyespots on the images. The algorithm is first trained with examples from a database of images with two different labels (eyespot and non-eyespot), and subsequently is able to provide classification for a new image. After an eyespot is detected the radius measurements of its color rings are performed by a 1D Hough Transform which corresponds to histogramming. We trained software to recognize eyespot patterns of the nymphalid butterfly *Bicyclus anynana* but eyespots of other butterfly species were also successfully detected by the software.

1. Introduction

Work in behavioral ecology, in evolutionary and developmental biology and in quantitative and functional genetics of butterfly wing patterns has been accelerating in recent years but researchers in these fields still lack an efficient measuring tool to quantify wing pattern variation. Until now most quantifications of wing pattern variability performed in butterflies are done by mouse-clicking landmarks around each of the targeted wing pattern elements, either using an image analysis software and a photograph of each animal, or using the live animal positioned under a microscope with a camera lucida attachment and clicking on a digitizing pad. Due to the slowness of manually scoring wing patterns, in experiments where hundreds of individuals must be scored, only a few of these wing patterns are usually measured in each individual, and analysis of the complete set of patterns are seldomly done.

Most of the research on quantitative genetics of butterfly wing patterns has focused on the circular eyespot patterns that are present along the border on the wing in a variety of nymphalid butterflies including two main model species, the buckeye, *Junonia (Precis) coenia*, and the squinting bush brown, *Bicyclus anynana*. Other nymphalids, however, such as *Heliconius* species (Joron et al., 2006), and the specked wood, *Pararge aegeria* (Breuker et al., 2007), are also targets of similar large-scale quantitative approaches. Research questions regarding the eyespot patterns have ranged from (1) quantifying the effect of environmental temperature and other variables on changes in the size of the eyespots (Kooi et al., 1996; Roskam and Brakefield, 1996); (2) determining patterns of eyespot covariation (Allen, 2007; Paulsen and Nijhout, 1993; Monteiro et al., 1994, 1997; Beldade and Brakefield, 2003); (3) discovering the extent to which each of these eyespots is free to vary independently of the others by using artificial selection or mutagenesis experiments (Beldade et al., 2002; Monteiro et al., 2003); (4) discovering which genes underlie eyespot pattern variation via linkage association studies (Monteiro et al., 2007; Beldade et al., 2002); (5) estimating the effect of ectopic expression or knockdown of candidate developmental genes on eyespot morphology (Monteiro and Chen, in prep.); (6) and understanding the evolution of eyespot number (Monteiro, in press). In order to accelerate the pace of investigation on butterfly quantitative and functional genetics we developed software that automatically recognizes and measures several of the color rings in each of the eyespot patterns using digital photographs of wings. We first trained and tested the software on a set of images from dissected wings of *B. anynana* but later tested the software on images from other *Bicyclus* species to estimate its flexibility in recognizing eyespot patterns in general. This software was specifically developed to recognize circular patterns on images and can potentially be of broader applicability within or outside the biological sciences. The main goals for the software were to (1) recognize all eyespot pattern elements on an image and count the number of eyespots on each wing surface, and (2) measure the radius of the different color rings in each eyespot.

* Corresponding author. Tel.: +351 218418297.
E-mail address: msilveira@isr.ist.utl.pt (M. Silveira).

0303-2647/$ – see front matter © 2008 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.biosystems.2008.09.004

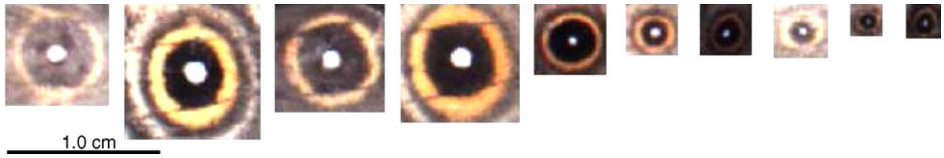

Fig. 1. Diversity of eyespots patterns in *Bicyclus anynana*. Different eyespots have different sizes, number of rings, color and contrast.

2. Materials and methods

The automatic eyespot recognition software was developed in Matlab 7.1, using the Image Processing Toolbox and the Statistical Pattern Recognition Toolbox and is made available at http://www.isr.ist.utl.pt/ msilveira/eyespot_recognition.htm.

2.1. The Butterflies

The wings of 250 *B. anynana* individuals, reared at 28 ° C and 80% humidity, were separated from the body and photographed using a Nikon SMZ1500 dissecting microscope at 3.8× magnification and a digital camera (Qimaging Micropublisher RTV). The wings were lit from left and right sides with a fiber optic double gooseneck connected to a cold light source. The photos were taken at 300 dpi and are 5.74 cm × 4.33 cm in size; they were saved as Tiff files.

2.2. Automatic Recognition Software

Our approach for the automatic eyespot recognition was to train a machine learning algorithm which assigned one of two possible labels (eyespot or non-eyespot) to each image pixel based on measurements obtained in the neighborhood of that pixel. Those measurements were collected into a feature vector x of length n and the algorithm output was based on a discriminant function $g(x)$ that partitions the feature space $\mathbb{R}^n$ into two decision regions:

$$g(x) = w^T x + b \quad (1)$$

where $b \in \mathbb{R}$ and $w \in \mathbb{R}^n$ are the coefficients of the linear discriminant function which had to be learned from examples of images with both labels (eyespot and non-eyespot). These example images corresponded to smaller square areas of the original digital images, including individual eyespots (Fig. 1) or background wing patterns. The background images were randomly generated throughout the wings in order to capture the diversity of the wing texture. The function yields positive values for eyespot examples and negative values for non-eyespot. Thus, training examples from the two different classes are separated by the hyperplane $g(x) = w^T x + b = 0$.

2.2.1. Features

The eyespot patterns are approximately circular and formed by concentric rings, but they have different sizes, number of rings, brightness values and contrast (Fig. 1). In the particular case of *B. anynana*, one of the rings in the eyespots has a distinctive gold color that could be used as a feature but we refrained from using color features because we intend to use the software with other species.

The features we used exploit the fact that the eyespots are circular and symmetric relative to their center. We obtained good results with a very reduced number of features, which were carefully selected. One set of features measures circularity and is inspired by the convergence index filter (Kobatake and Hashimoto, 1999) which was designed to detect rounded regions. This filter measures the degree of convergence of gradient vectors in the neighborhood of the pixel of interest. Let p denote the center pixel of a region R and q denote an arbitrary pixel in R with relative coordinates from the center pixel $q = (k, l)$. The gradient of image $I(q)$ is denoted $g(q) = (g_x(q), g_y(q))$. From $g_x(q)$ and $g_y(q)$ the gradient magnitude and orientation can be calculated:

$$\|g(q)\| = \sqrt{g_x(q)^2 + g_y(q)^2} \quad (2)$$

$$\phi(q) = \arctan \frac{g_y(q)}{g_x(q)} \quad (3)$$

The angle $\theta(q)$ measures the orientation of the gradient vector $g(q)$ with respect to the line $\overline{pq}$ and the degree of convergence of $g(q)$ is given by $\cos\theta$ (Fig. 2a). The convergence index output is the average of the convergence indices at all pixels in R:

$$c(p) = \frac{1}{M} \sum_{q \in R} \cos\theta(q) \quad (4)$$

where M is the number of pixels in region R. We adapted the convergence index filter because in our case, the eyespots have both dark and light rings, so some gradient vectors will point towards the pixel of interest and others will point away from it. In the first case the values of $c(q)$ will be positive and in the second case they will be negative. Therefore, we divided the pixels in region R into two sets based on their angle $\theta(q)$:

$$R^+ = \{q \in R | \cos\theta(q) \geq 0\} \quad (5)$$

$$R^- = \{q \in R | \cos\theta(q) < 0\} \quad (6)$$

We calculated $\cos\theta(q)$ efficiently by using the following normalized dot product:

$$\cos\theta(q) = \frac{g(q).v(q)}{\|g(q)\|\|v(q)\|} \quad (7)$$

where v is the vector $q - p$.

The filter output is multiplied by the gradient magnitude to give more weight to the more contrasted points. This is done because gradient elements with small magnitude have less reliable orientation. In addition, the output is scaled by the total gradient magnitude in order to obtain a measure adapted to local contrast:

$$c^+(p) = \frac{\sum_{q \in R^+} \|g(q)\| \cos\theta(q)}{\sum_{q \in R^+} \|g(q)\|} \quad (8)$$

$$c^-(p) = \frac{\sum_{q \in R^-} \|g(q)\| \cos\theta(q)}{\sum_{q \in R^-} \|g(q)\|} \quad (9)$$

Another feature measures radial gradient and was used in Daugman (2004) to localize and recognize a human iris. It is an integrodifferential operator that calculates at center coordinates, p, in the image domain, the blurred partial derivative with respect to increasing radius, r, of the normalized contour integral of I along a circular arc ds of radius r:

$$rg(r, p) = G_\sigma * \frac{\partial}{\partial r} \int_{r,p} \frac{I(q)}{2\pi r} ds \quad (10)$$

The symbol $*$ denotes convolution and G_σ is a smoothing function such as a Gaussian of scale σ. As our feature we use the average of the radial gradient $rg(r, p)$ computed for all values of the radii r.

Two additional features exploit the pattern's gradient symmetry relative to the center point. Using symmetry as a feature is important to avoid false detections from the curved chevron patterns present along the border of the wing. Moreover, this symmetry feature is useful to detect eyespots with elliptic shapes. We used one of the symmetry features proposed in Loy and Zelinsky (2003) but calculated dark and bright symmetry separately.

We calculate at each radius r two projection images O_r+ and O_r-, that will collect evidence of dark and bright symmetry, respectively. To create these images, for each point p we calculate the pixel p_+ that the gradient vector $g(p)$ is pointing

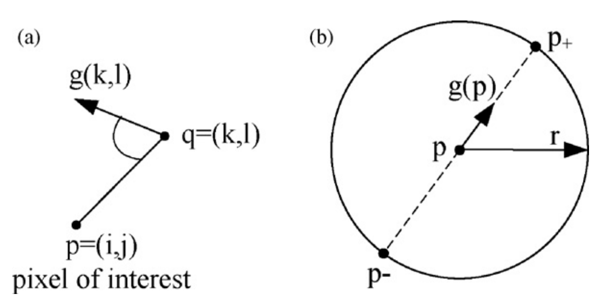

Fig. 2. Gradients and angles used in the circularity and symmetry features. (a) circularity; (b) symmetry.

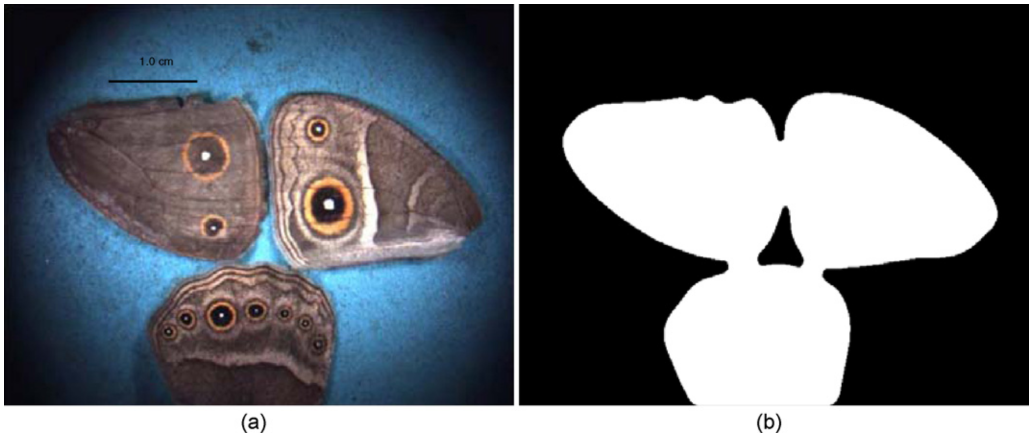

Fig. 3. Example of wing/background separation. Only the pixels shown in white are scanned. (a) Original image; (b) corresponding wings/background mask.

to, a distance r away from p and the pixel p_- that the gradient is pointing directly away from, as shown in fig. 2 b. The coordinates of these pixels are given by

$$p_+(p) = p + round\left(\frac{g(p)}{\left|h(p)\right|}r\right) \tag{11}$$

and

$$p_-(p) = p - round\left(\frac{g(p)}{\left|g(p)\right|}r\right) \tag{12}$$

where '*round*' rounds each coordinate to the nearest integer.

The projection images are initially zero. Then, O_r+ is increased by 1 at all points $p+$ while O_r- is decreased by 1 at all points $p-$:

$$O_{r+}(p_+(p)) = O_{r+}(p_+(p)) + 1 \tag{13}$$

$$O_{r-}(p_-(p)) = O_{r-}(p_-(p)) - 1 \tag{14}$$

The symmetry contribution, either bright or dark, at radius r is defined as the convolution:

$$S_r = F_r * A_r \tag{15}$$

where

$$F_r = \mathrm{sgn}(\tilde{O}_r(p))\left(\frac{\tilde{O}_r(p)}{k_r}\right)^{\alpha} \tag{16}$$

and

$$\tilde{O}_r(p) = \begin{cases} O_r(p) & \text{if} O_r(p) < k_r \\ k_r & \text{otherwise} \end{cases} \tag{17}$$

In this equation O_r stands for either O_r+ or O_r-, A_r is a two-dimensional Gaussian, α is the radial strictness parameter, and k_r is a scaling factor that normalizes O_r across different radii. This result is contrast independent. However, when applying this orientation-based formulation we ignore very small gradients that tend to add noise to the result. More details can be found in Loy and Zelinsky (2003).

These features are calculated at every candidate image pixel p and with different sizes for region R. We used square regions with 31×31, 51×51 and 71×71 pixels.

2.2.2. Machine Learning Algorithm

We used a support vector machine (SVM) (Vapnik, 1998) classifier using the features described above. This classifier finds the separation hyperplane that maximizes the separation margin between the two classes (eyespot and non-eyespot) and has proved to be one of the best in terms of generalization ability. The hyperplane is found by minimizing the following cost function:

$$f(w) = \frac{1}{2}\|w\|^2 \tag{18}$$

subject to the constraints: $w^T x_i + b \geq 1$ for the positive examples, and $w^T x_i + b \leq -1$ for the negative ones. The solution will only depend on a subset of the training examples which are the support vectors.

Since in practice the training examples may not be completely separated by a hyperplane, slack variables $\xi_i \geq 0$ can be introduced to relax the separability constraint:

$$w^T x_i + b \geq 1 - \xi_i \tag{19}$$

Accordingly, the cost function becomes:

$$f(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{P}\xi_i \tag{20}$$

where C is a user defined, positive regularization parameter. Increasing the value of C increases the cost of misclassifying training examples and creates a more accurate model that may not generalize well. In all our experiments we used $C = 1$.

Other more common approaches for circle detection such as the Hough Transform (HT) (Duda and Hart, 1972) or even HT variants especially designed for concentric circle detection (Silveira, 2005) were found unsuitable because many eyespots are very close to each other which generated numerous false peaks in the HT center accumulator in the regions between neighboring eyespots.

2.2.3. Preprocessing

The number, position and also size of the eyespots is variable so the images have to be scanned not only at different locations but also at different scales. To reduce computation, the wing region was separated from the background prior to scanning. This separation was based on the hue values of the image. Noting that the background was blue for all the images in our data set we extracted image pixels with hue values between 0.5 and 0.6, which correspond to blue colors. After that, a morphological flood fill operation was performed on the binary image in order to eliminate holes in the wings.

2.2.4. Postprocessing

The method described above is not sensitive enough to small changes in eyespot location or size. Therefore, multiple detections will usually occur around each eyespot in a scanned image and the same will happen for false detections. In order to reduce this effect we performed non-maxima suppression using the value of the discriminant function $g(x)$ at neighboring scales and locations. This operation suppresses all detections except for the ones corresponding to local maxima of the discriminant function.

2.3. Hand Measurements

In order to compare the performance of our automatic eyespot detection and measuring software with detections done by eye and measurements done by hand, we scored the total number of eyespots present on each of the 250 images by eye and measured several eyespot diameters using Object Image 1.62 software (Vischer et al., 1994). This software allows users to obtain linear measurements (in this case eyespot diameters) by calculating the distance between two xy coordinates that were "mouse clicked" on a Tiff image by hand. Ten linear measurements were obtained on all forewings: the diameter of the black disc and gold ring for both the anterior and posterior eyespots on the dorsal and ventral surfaces, and the diameter of the white pupils of the posterior eyespots on the dorsal and ventral surfaces. In addition, we obtained x, y coordinates for the center of each eyespot.

2.4. Automatic Scoring

We developed a method to compare the automatically detected eyespots with the manually scored eyespots. For each manually scored eyespot we identified the best matching automatically detected eyespot by comparing first their x,y center coordinates and then by testing whether the area of overlap of the color rings was greater than a fixed threshold T. Detections that were not matched to manually scored eyespot were counted as false positives, and detections that matched manually scored eyespots (via their x, y coordinates) but where the overlap of area

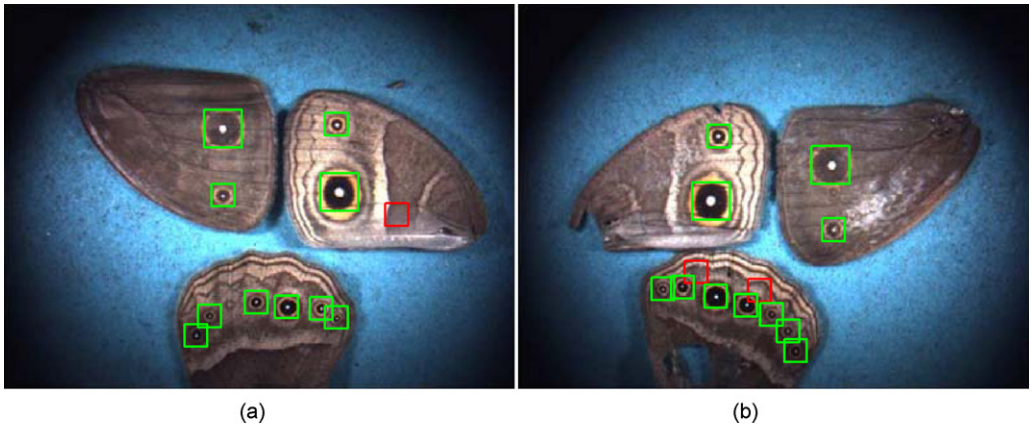

Fig. 4. Some examples of detection results on *B. anynana*. Green squares indicate correct detections whereas red squares indicate false positives. (a) The software detected 11 eyespots, 10 were true eyespots and one detection was a false positive. In addition, there was one missed detection on the ventral surface of the hindwing. (b) The false positives occur mainly along the border of the wing where curved chevron patterns are present.

measurements were below the set threshold were also counted as false positive. Manually scored eyespots that were not matched with any detected eyespot were counted as missed detections.

We used 2500 examples of eyespot images (Fig. 1) and generated 25,000 examples of non-eyespot images to train the recognition algorithm. The non-eyespot images were randomly generated throughout the wings, after removing the background portion of the images (Fig. 3).

The proposed classifier was evaluated using twofold cross-validation; the database of images was divided in a random fashion into two subsets of the same size, where each subset was used in turn for training and the other for testing. We calculated the overall true detection rate (TDR) and false positive rate (FPR) as follows:

$$\mathrm{TDR} = \frac{\mathrm{TD}}{\mathrm{GT}} \tag{21}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TD}} \tag{22}$$

where TD is the number of true detections, FP is the number of false positives and GT is the total number of eyespots in the database of images.

2.5. Eyespot Measurements

After an eyespot is recognized in the image, we proceeded to measure the radii of its different color rings. First we obtained the edge points in the eyespot region with the canny edge detector (Jain, 1989) and for each edge point $e = (x, y)$ we computed its radius relative to the center point $p = (i, j)$ using the circle equation:

$$r^2 = (x - i)^2 + (y - j)^2 \tag{23}$$

We then constructed an accumulator for the radius where each edge point casts a vote on the value of its radius. Concentric circles of different radii (e.g., the white pupil, the black disc, the gold ring) will originate different peaks. In order to obtain sharper peaks, the votes are weighted by the value of the circularity features described above (see Eqs. (8) and (9)). In addition, since larger concentric circles will also originate higher peaks in the radius accumulator, each count is divided by the corresponding radius value.

In order to compare the manual and the automatic eyespot measurements we calculated the error by subtracting the linear measurements from the automatic measurements and averaging the differences. We tested whether there was a significance difference between the two measurements using a Wilcoxon paired signed-rank test.

3. Results

After training the algorithm we evaluated its performance in recognizing an eyespot with the automatic scoring method described in Section 2.4. The true detection rate was 96% with 6% of all detections being false positives (see example detections in Fig. 4). Most of the false positives occur in the area of the wing periphery where there are concentric arc ring patterns, the “chevrons”, in butterfly wing pattern nomenclature (Fig. 4b). Most of the missed detections correspond to the smaller eyespots. In fact, if we disregard eyespots smaller than 25×25 pixels, our recognition rate increases to 98% with 8% false positives. The localization error of the center of the eyespots that were correctly identified was 1.1 pixels. The application of the non-maxima suppression described in Section 2.2.4 reduced the number of overlapping detections of the same eyespot to a single detection. This method also reduced the number of false positives (Fig. 5). Using methods based on the Hough Transform,

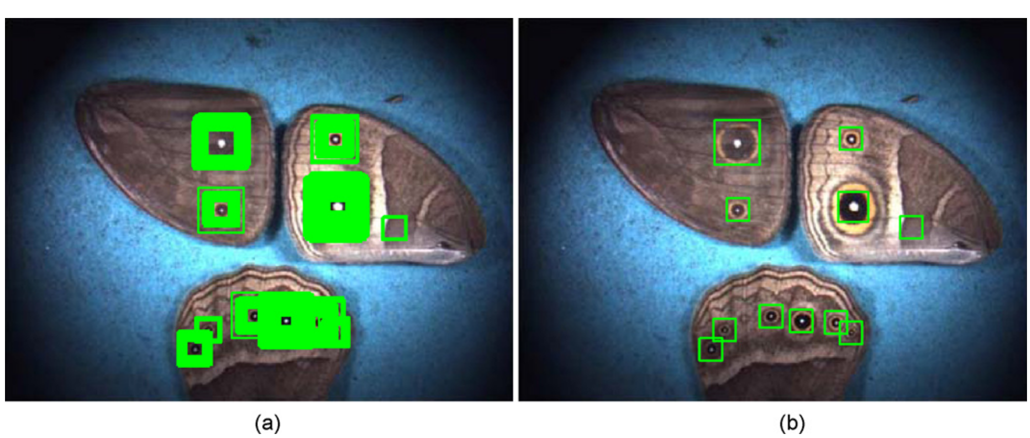

Fig. 5. The effect of non-maxima suppression. (a) Before non-maxima suppression; (b) after non-maxima suppression.

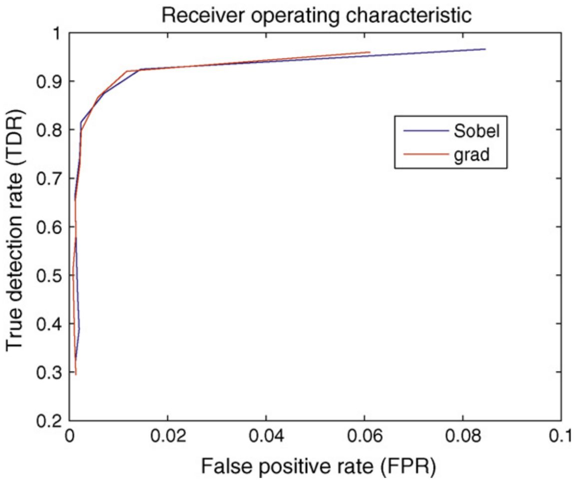

Fig. 6. Receiver operating characteristic (ROC) curves of recognition performance. The curve labeled *grad* corresponds to the gradient calculated with horizontal and vertical central differences and the curve labeled *Sobel* was produced using the Sobel operator (Jain, 1989).

we achieved detection rates of 100% but with unacceptable false positive rates, in all cases around 100%.

We also calculated receiver operating characteristic (ROC) curves showing the tradeoff between probability of true detection and false detections, obtained with varying classifier threshold b (see Eq. 1). Since all our features are based on gradient direction, we compared the ROC curves of our algorithm when two different methods were used to calculate the gradient (Fig. 6). One method used the gradient calculated with horizontal and vertical central differences and the other one used the Sobel operator (Jain, 1989). In these ROC curves the false positive rate never reaches 1.0 because of the non-maxima suppression procedure described in Section 2.2.4. These results indicate that the performance of our method is not very sensitive to the way the gradient was calculated, although the gradient calculated with horizontal and vertical differences was slightly superior.

Finally, in order to evaluate the generalization ability of our software to detect eyespots, we tested it on images of other butterflies species. Forty five images of different species with eyespots quite different from *B. anynana* such as *B. dentatus*, *B. safitza*, *B. asochis*, *B. hyperanthus*, *B. angulosus* among several others were tested (Fig. 7). The detection rate was 67% and the false positive rate was 13%. Naturally, these results are inferior to the ones obtained when the eyespots of *B. anynana* were used both for training and for testing the SVM classifier, but they are very promising especially since we

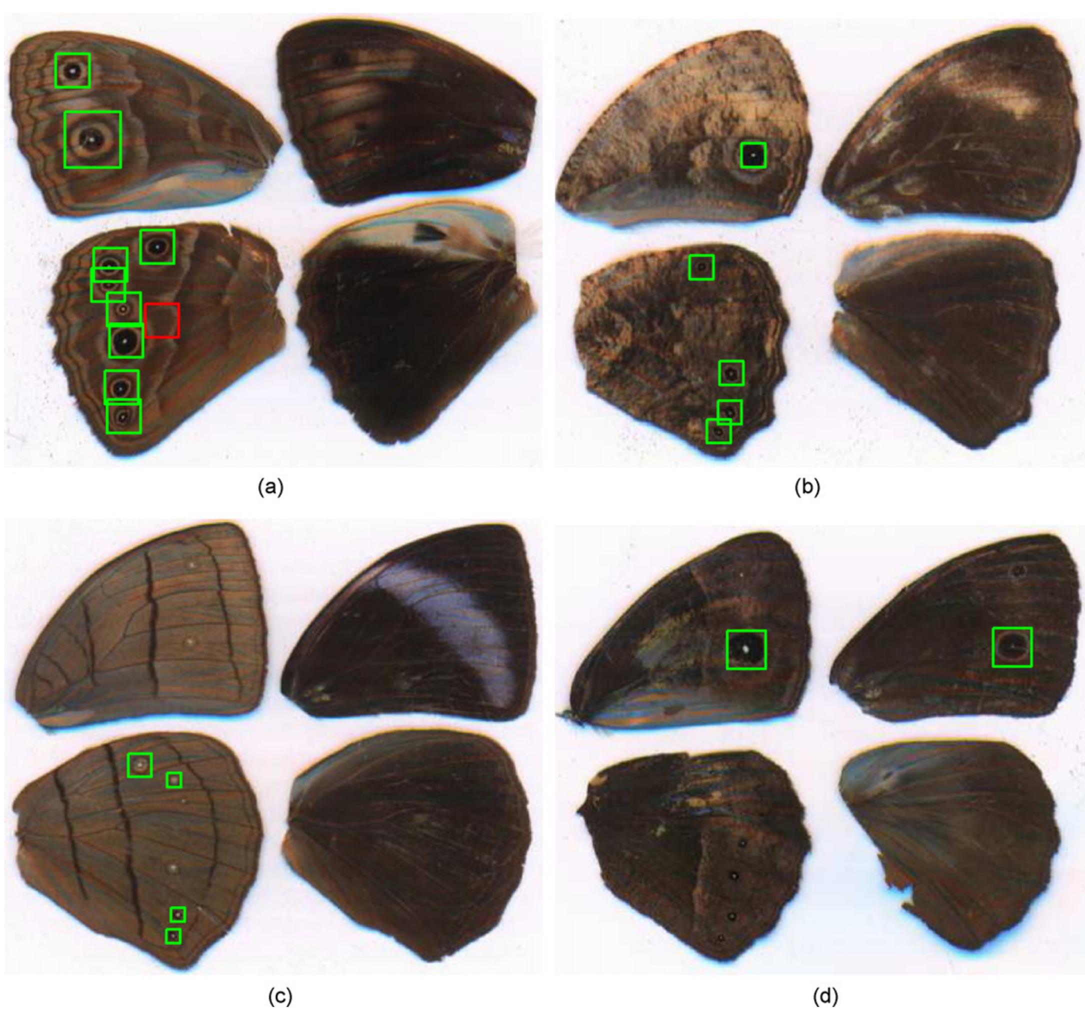

Fig. 7. Detection results obtained with different *Bicyclus* species other than *B. anynana*. (a) *B. xeneoides*, (b) *B. auricrudus*, (c) *B. iccius*, and (d) *B. safitza*.

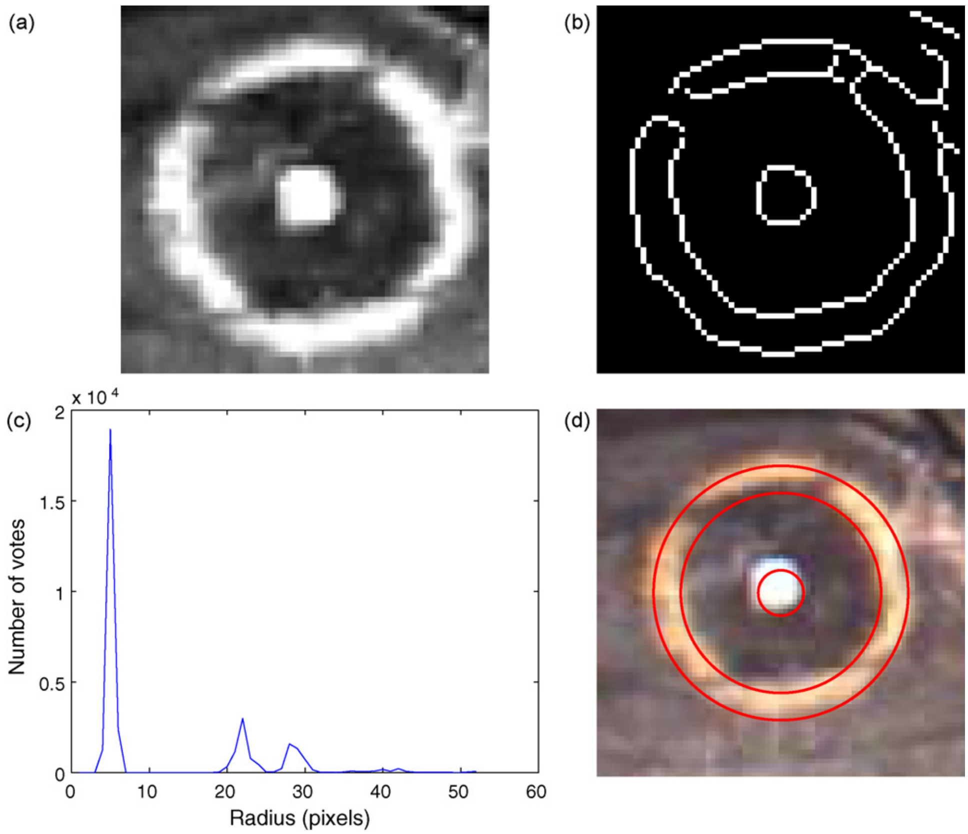

Fig. 8. Eyespot measurements. (a) Luminance image of an eyespot; (b) corresponding edge map; (c) corresponding radius accumulator; (d) the measured circles are superimposed in red. Three maxima appear corresponding to three concentric circles. These maxima map to the edges of the different color rings. The strongest peak represents the edge of the central white pupil. The identified circles were $r = 5$, $r = 22$ and $r = 28$ pixels in radius.

used so few and such simple features to design our automatic eyespot recognition algorithm. In order to get the best performance, the algorithm should be tested and trained with pictures of the same butterfly species.

The training step took around 10 min and testing and measuring took approximately 3.5 min per image, running on a conventional laptop (2GB RAM, Dual Core CPU, 2 GHz). These measurements, however, can be done automatically without input from the user. Measurements by hand take approximately one minute and thirty seconds for each specimen, for a total of 11 eyespots and 3 color rings in each eyespot. The number of clicks per image is 66, which is rather large and can potentially lead to repetitive strain injury. In addition, time savings can be achieved in the future by implementing the current Matlab code in a different language, e.g. C, lending to faster automatic detections and measurements.

3.1. *Eyespot Measurement Results*

The edges of the correctly detected eyespots in the *B. anynana* images were used to build a Hough Transform accumulator where the radii of its color rings were measured (Fig. 8). We compared the manual measurements of the eyespot diameters with the automatic ones and found very small differences between the two (Table 1).

Table 1
Average error and error standard deviation (both in pixels) of manually and automatically measured eyespot color radii.

	Error	Std. Dev.
White pupil	−0.81	0.95
Black disk	0.24	2.12
Gold ring	0.02	2.52

Most of the differences between the manual and the automatic measurements are due to the assumption that the three circles are concentric which is just an approximation. In fact, many eyespots are elliptic rather than circular. The greater error standard deviation obtained for the gold ring is due to this being the most elliptic of the rings (the outer ring), and also the one displaying the least contrast with the background color immediately following its outer edge.

We tested whether there was a significant difference between radius measurements when done by hand or automatically using a Wilcoxon paired sign-rank test. While the white pupils measured by hand were significantly larger than those measured automatically ($Z = -10.44; p = 0.000$), the measurements for the black and gold ring were comparable (black: $Z = -1.25$, $p =$ ns;gold : $Z = -0.186$, $p =$ ns). Our interpretation of these results is that the automatic pupil measurements are more accurate as it is difficult to precisely click on the pupil coordinates by hand, and apparently we bias our manual measurements by “enlarging” the pupil diameters.

4. Conclusions

Our automatic eyespot recognition software proved to recognize and identify around 96% of the total number of eyespots present on images of the dorsal and ventral wing surfaces of the butterfly *B. anynana*. Furthermore the software recognized the outlines for the white pupil, the black disc, and the gold ring of scales in each eyespot, across a range of sizes. The software also performed well in dealing with eyespot variation in outline crispness, brightness, and contrast levels with background pattern. The radii measurements done on the basis of the color ring outlines are comparable to the manual diameter measurements. Future developments will include the implementation of this software in a user-friendly web interface and the inclusion of features that will allow users to correct the false positive detections and manually point to eyespots that were not detected by the software. In order to scale for overall wing size, the user interface will also include the ability to obtain linear measurements between two mouse clicked *xy* coordinates or to obtain automatic wing area measurements. Once these features are implemented, this software is likely to speed the pace of research on the evolution and development of eyespot patterns in butterfly wings.

Acknowledgments

We thank Diane Ramos for the images used in this study and Gershon Dublon for manually scoring the eyespots. We also thank Peter Kovesi for the publicly available implementation of Loy and Zelinski's fast radial symmetry detector. This work was supported by Fundação para a Ciência e a Tecnologia (ISR/IST plurianual funding) through the POS Conhecimento Program that includes FEDER funds and by National Science Foundation Research Experience for Undergraduates (NSF REU) supplement to award IOB-0653399/D01357 to AM for supporting GD. We are grateful to the anonymous reviewers for their helpful comments and suggestions.

References

Allen, C.E., 2007. The "Eyespot Module" and eyespots as modules: development, evolution, and integration of a complex phenotype. J. Exp. Zool. B Mol. Dev. Evol. 310, 179-190..

Beldade, P., Koops, K., Brakefield, P.M., 2002. Developmental constraints versus flexibility in morphological evolution. Nature 416, 844–847.

Beldade, P., Brakefield, P.M., 2003. Concerted evolution and developmental integration in modular butterfly wing patterns. Evol. Dev. 5, 169–179.

Beldade, P., Brakefield, P.M., Long, A.D., 2002. Contribution of Distal-less to quantitative variation in butterfly eyespots. Nature 415, 315–317.

Breuker, C.J., Gibbs, G.W., Van Dyck, H., Brakefield, P.M., Klingenberg, C.P., Van Dongen, S., 2007. Integration of wings and their eyespots in the speckled wood butterfly *Pararge aegeria*. Mol. Dev. Evol. B 308, 454–463.

Daugman, J., 2004. How iris recognition works. IEEE Trans. Circuits Syst. Video Technol. 14 (January (1)), 21–30.

Duda, R., Hart, P., 1972. Use of the Hough Transformation to detect lines and curves in pictures. Commun. ACM 15 (1), 11–15.

Jain, A.K., 1989. Fundamentals of digital image processing Prentice Hall.

Joron, M., Jiggins, C.D., Papanicolaou, A., McMillan, W.O., 2006. Heliconius wing patterns: an evo-devo model for understanding phenotypic diversity. Heredity 97, 157–167, Prentice Hall.

Kobatake, H., Hashimoto, S., 1999. Convergence index filter for vector fields. IEEE Trans. Image Process. 8, 1029–1038.

Kooi, R.E., Brakefield, P.M., Rossie, W.E.M.-T., 1996. Effects of food plant on phenotypic plasticity in the tropical butterfly *Bicyclus anynana*. Entomol. Exp. Appl. 80, 149–151.

Loy, G., Zelinsky, A., 2003. Fast radial symmetry for detecting points of interest. IEEE Trans. Pattern Anal. Mach. Intelligence 25 (8), 959–973.

Monteiro, A.F., Brakefield, P.M., French, V., 1994. The evolutionary genetics and developmental basis of wing pattern variation in the butterfly *Bicyclus anynana*. Evolution 48, 1147–1157.

Monteiro, A., Brakefield, P.M., French, V., 1997. Butterfly eyespots: the genetics and development of the color rings. Evolution 51, 1207–1216.

Monteiro, A., Prijs, J., Bax, M., Hakkaart, T., Brakefield, P.M., 2003. Mutants highlight the modular control of butterfly eyespot patterns. Evol. Dev. 5, 180–187.

Monteiro, A., Chen, B., Scott, L., Vedder, L., Prijs, J.H., Belicha, A., Brakefield, P.M., 2007. The combined effect of two mutations that alter serially homologous color pattern elements on the fore and hindwings of a butterfly. BMC Dev. Biol. 8, 22.

Monteiro, A., 2008. Alternative models for the evolution of eyespots and serial homology on lepidopteran wings. Bioessays 30, 358-366.

Paulsen, S.M., Nijhout, H.F., 1993. Phenotypic correlation structure among elements of the color pattern in *Precis coenia* (Lepidoptera: Nymphalidae). Evolution 47, 593–618.

Roskam, J.C., Brakefield, P.M., 1996. A comparison of temperature-induced polyphenism in African Bicyclus butterflies from a savannah-rainforest ecotone. Evolution 50, 2360–2372.

Silveira, M., 2005. An algorithm for the detection of multiple concentric circles. Lecture Notes Comp. Sci. Pattern Recogn. Image Anal. 35223, 271–278.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.

Vischer, N.O.E., Huls, P.G., Woldringh, C.L., 1994. Object-Image: an interactive image analysis program using structured point collection. Binary (Bioline), 6.