

Detection, Counting, and Classification of Visual Ganglia Columns of *Drosophila* Pupae

Enrique Javier Arriaga Varela^{1,2}, Eduardo Ulises Moya Sánchez^{1,2}, Armando Aguilar Meléndez^{1,3}, Octavio Castillo Reyes¹, Eduardo Vazquez Santacruz⁶, Sebastian Salazar Colores⁴, Ulises Cortés^{1,5}

¹ Barcelona Supercomputing Center (BSC), Barcelona, Spain

² Universidad Autónoma de Guadalajara, Guadalajara, Mexico

³ Universidad Veracruzana, Poza Rica, Mexico

⁴ Universidad Autónoma de Queretaro, Queretaro, Mexico

⁵ Universitat Politècnica de Catalunya - BarcelonaTECH, Barcelona, Spain

⁶ Universidad Autónoma del Carmen, Ciudad del Carmén, Mexico

eduardo.moyasanchez@bsc.es

Abstract. Many neurobiologists use the fruit fly (*Drosophila*) as a model to study neuron interaction and neuron organization and then extrapolate this knowledge to the nature of human neurological disorders. Recently, the fluorescence microscopy images of fruit-fly neurons are commonly used, because of the high contrast. However, the detection of the neurons or cells is compromised by background signals, generating fuzzy boundaries. As a result, it is still common that in many laboratories, the detection, counting, and analysis of this microscope imagery is still a manual task. An automated detection, counting, and morphological analysis of these images can provide faster data processing and easier access to new information. The main objective of this work is to present a semi-automatic detection-counting system and give the main characteristics of images of the visual ganglia columns in *Drosophila*. We present the semi-automatic detection, count, segmentation and we concluded that it is possible to obtain an accuracy of 75% (with a Kappa statistic of 0.50) in the shape classification.

Additionally, we develop python GUI CC Analyzer which can be used by neurobiology laboratories whose research interests are focused on this topic.

Keywords. Image processing, computer vision, machine learning, fruit fly (*Drosophila*), visual ganglia columns.

1 Introduction

Many neurobiologists use the visual system of the fruit fly (*Drosophila*) as a model to study neuron interaction [4, 6, 9] and neuron organization into columnar units in the brain and extrapolate this knowledge to human behavior and diseases [5, 4, 8]. *Drosophila* studies have a significant value for the scientific community with five Nobel prizes related, because of working and manipulating the genome of the fruit fly is an easily reproducible

technique and a big amount of genetic tools are available. Additionally, this animal shows a relatively complex behavioral repertoire in a relatively simply organized brain, with only 100,000 neurons. Within the fly, the optic lobe is especially appealing because flies are highly visual animals, with the two optic lobes encompassing approximately 50% of the total brain volume.

The fluorescence microscopy images of columns commonly have high contrast. However, the detection is compromised by background signals, generating fuzzy boundaries of columns. Additionally, the number of elements are approximately 10^3 [6]. Therefore, the detection and counting of columns are still difficult due to the manual analysis nature that involves the technique. An automated detection, counting, segmentation and morphologic analysis of columns can provide to researchers a significant advantage on speed and accuracy in the analysis of this kind of images.

The main aim of this work is to develop an algorithm for real-time shape recognition, counting, and classification of the *Drosophila* pupa visual ganglia column images obtained by using a confocal microscope. The main contribution of this algorithm is that we combine automatic operations with image analysis results and user parameters in real-time.

This paper is divided into four sections. In the next section, we present the methods and data for the image processing algorithm (including its limitations), detection, segmentation, and the decision tree algorithm to obtain the rules which define a set of good shape or bad shape columns. In section 3, we present our images analysis results. Finally, in section 4, we provide conclusions.

2 Data and Methods

In this section we describe the data and propose a real-time semi-automatic image processing algorithm to detect, visualize, find edges, compute the position (centroid) for fluorescence of optic lobule fly pupae images.

2.1 Data

We have used 12 fluorescence microscopy images (120 segmented shapes, 8bit-RGB, JPEG images of 500x500 pixels). Figure 1 shows an example of original images.

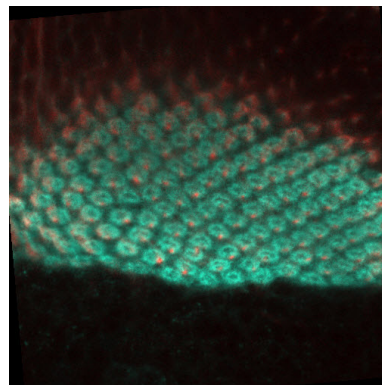


Fig. 1. Example of fluorescence microscopy image of fly pupae

2.2 Image Processing

We have used Open CV 3.0 [7] with Python to develop our real-time image processing and analysis of a selected Region of Interest (ROI) selected by the user, with a fixed size 150 x 150 pixels.

The algorithm to detect and segment cell-shapes is described as follows:

1. RGB image is converted into gray 8-bit grayscale.
2. Bilateral filter is applied to the gray image ROI [3]. This filter is highly effective in noise removal while keeping edges sharp.
3. An adaptive binary threshold operation is used, with an initial threshold C_0 value of -3. However, the user can define the C .
4. A dilate operation is applied to approximate the real size of the detected shapes.
5. The contours shapes are obtained from the binary image with the method [10].

6. Compute the proportion between width and height and compare with the main value of the other shapes.
7. The centroids are computed to *count the possible cells* and serves as a visual aid.
8. Draw the visual shape over the first copy of the squared image (as shown in Fig 4).

The main contribution of this image preprocessing algorithm is that, we can combine automatic image processing with image analysis, and user parameters to fine-tune the results and display them in real-time. It is possible to see and download this code¹.

2.3 Image Processing Limitations

Due to the extremely variable quality of the microscopic image, even at same the anatomic region, we measure the range of Signal to Noise Ratio (SNR), where our algorithm is reliable. To measure the SNR we used the mean value of the pixel intensity L as a signal and variance as a noise (of a normal distribution) of some Regions of Interest (ROI) with gray images. The SNR can be obtained as follows:

$$SNR = \frac{Signal}{Noise} = \frac{mean(L)_{ROI}}{var(L)_{ROI}}. \quad (1)$$

In the case that the image has a low value of SNR the user has the option to select or remove detected shapes individually as necessary.

2.4 Image Analysis and Feature Extraction

The image analysis computation is focused in two main areas compute the number of possible shapes of the column and obtaining features for their classification. We use the expertise of the laboratory team and our experimental results in order to define the main features. The main features were the width, height, perimeter, area, the centroid of each segmented shape. In addition, the main shape analysis measurements and operations are:

1. Proportion between width and height: WHPROPORTION,
2. Proportion between perimeter and area: PXA,
3. Average changes of color per row: CHANGESX,
4. Average changes of color per column: CHANGESY.

2.5 Cell Classification

We have segmented 120 shape cells, and we obtain two labels GOOD and BAD cells.

In order to do the image cell classification, we use the shape as a principal feature. We used a decision tree not only to obtain a prediction or classification result but also to obtain logical rules with a simple explanation. These rules can be correlated with the features of the anatomical shapes, expressed by the experts. It is worth to mention that the decision tree can compute an accurate value, even with little data and it is not a black box algorithm [12].

To create the decision tree we used *Weka* [11, 12]. The JRip algorithm [2] was used to classify the good and bad cells, with 10 folds cross-validation. To create the training input information we propose the following input features:

1. Centroid for each shape: CENTROID,
2. Proportion between width and height: PROPORTION,
3. Proportion between perimeter and area: PXA,
4. Average changes of color per row: CHANGESX,
5. Average changes of color per column: CHANGESY.

¹<https://github.com/enriqueav/CCAnalyzer>

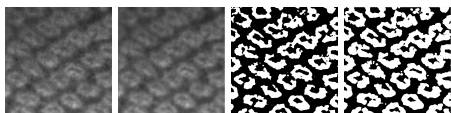


Fig. 2. From left to right: 8 bit gray scale ROI transform, bilateral filter application, adaptive threshold, and dilate operation

3 Results

3.1 Image Processing

The real-time image processing steps are presented in Figure 2. The RGB image is converted into 8-bit grayscale, then bilateral filter is applied to the gray image, after that, an adaptive binary threshold operation is used, finally, a dilate operation is applied.

The contours shapes from the binary image, we compute the centroids to count the number of cells and draw the visual shape over the first copy of the squared image (See Figure 3).

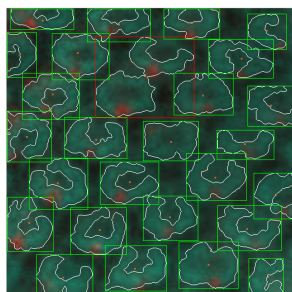


Fig. 3. Example of image detection, segmentation of a ROI

3.2 Image Processing Limitations

The SNR has been measured for each ROI. According to our measures, the proposed algorithm works better with $SNR > 4.7$, this value is closer to Rose criteria value [1].

In Figure 4 we show an example of the *ROI* of an image with $SNR = 3.6$, in this case, the algorithm has the ability to detect several shapes (white contours), but has some false positive results. Even with this SNR, the user can filter easily in the

application the bad shapes and not include them in the count.

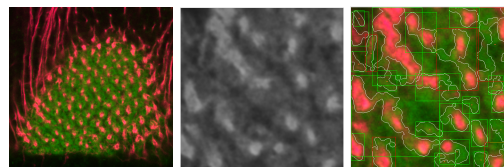


Fig. 4. Original image (left), ROI of image processed ROI image with $SNR = 3.67$ (center), and result of processing at the ROI (right)

Figure 5 shows the worst quality image that we have in our experiments. In this case the $SNR = 2.7$, and even in these conditions, our algorithm can detect some regular cell shapes.

However, the perimeter, area defined by each shape, is highly incorrect. We show the original image, the ROI, and the segmented images. Additionally, in this case, we present a plot profile with the intensity of the pixel L , it is possible to see that there almost the same signal of the cells and the background.

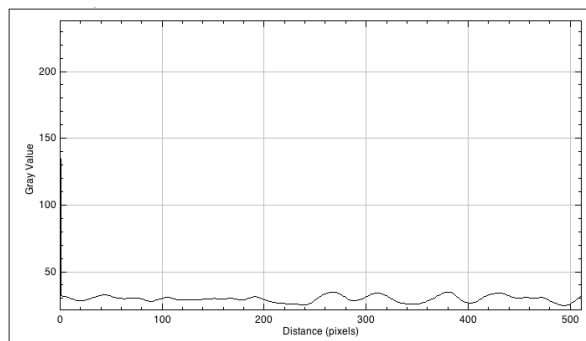
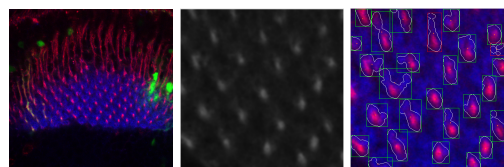


Fig. 5. Worst SNR image ($SNR = 2.7$) of the set. Original image (left), ROI(center), and the result of the segmentation (left). In the bottom part, we present a plot profile of the ROI

3.3 Image Analysis

In Table 1 we present the mean value and the standard deviation of features (F) $WHPROPORTION=W/H$, PXA , $CHANGESX=CHX$ and $CHANGESY=CHY$. of both classes (C) good (G) and bad (B). According to the results, the feature mean values and their standard deviation in both classes are crossing. As a consequence, is complicated to use of only a statistical method to define good cells from the bad ones.

Table 1. Image analysis results

F/C	W/H	PXA	CHX	CHY
G	1.4 ± 0.2	32.9 ± 12.2	3.3 ± 0.9	2.7 ± 0.6
B	1.5 ± 0.4	24.3 ± 7.0	4.0 ± 1.1	3.6 ± 1.1

Additionally, we count the number of cells and each position. We use $CENTROID$ to calculate how many cells are, their position in the image, and their classification. Moreover, we obtain some heuristic rules, that we used to improve the shape detection. The set of heuristic rules are:

1. Remove very small shapes, most likely created from remaining noise (perimeter < 30 pixels).
2. Remove the shapes that are totally enclosed by others, this is to avoid toroid-like shapes.
3. Remove the areas greater than 1.5 standard deviations (also show as a visual aid).
4. Remove all the incomplete forms and remove the shapes that more than 50% of the area is outside ROI.

Examples of good and bad shapes are shown in Figure 6. It is easy to see that the good cells are more rounded. In the case of the bad cells, their shape is irregular and speculated.

3.4 Shape Classification

The accuracy of the shape classification of the cells was 75%, with a Kappa statistic of 0.5031. The set of rules created are by the proposed decision-tree are:



Fig. 6. Subset of shapes used to train the classifier. Bad columns (top) and Good columns (below)

1. IF Perimeter and Area ($PXA \geq 33.42$) $ISGOOD=1$ (53.0/8.0).
2. IF average changes of color in Y ($CHANGESY \leq 3.27$) and proportion between width and height ($WHPROPORTION$ $1.46 \leq 1.63$) $ISGOOD = 1$ (15.0/2.0).
3. IF ($CHANGESY \leq 2.51$) $ISGOOD = 1$ (11.0/3.0).
4. ELSE $ISGOOD = 0$ (92.0/15.0).

The rules obtained are consistent with the expert criteria. However, in future work, we want to explore to a convolutional neural network in order to evaluate the classification of the shapes.

3.5 CC Analyzer GUI

The Figure 7 where is shown the shape, variables, the count of the cells and the result of the classification. It is possible to see the contour of the enclosing rectangles for each of the detected shape.

Rectangles are printed using 2 different colors: green when the shape is marked as good shape and red when the shape is marked as bad shape.

Additionally, the computed cell shape is presented in white and the centroid is drawing in orange. We believe that this GUI can be useful to create a bigger dataset in order to improve the shape classification.

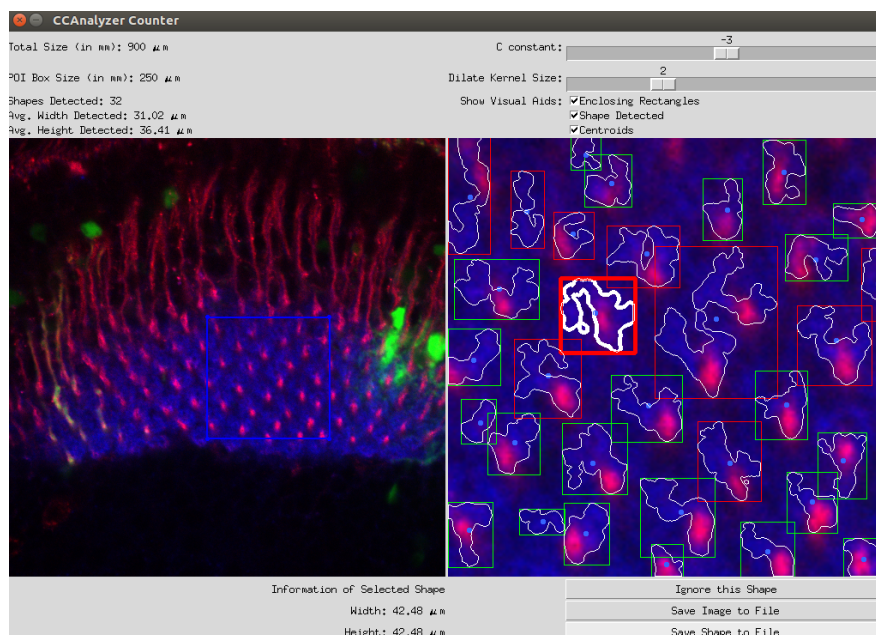


Fig. 7. The GUI and the result of the final classification of a ROI

4 Conclusions

We have presented our real-time algorithms to detect, count and classify correct and disorganized columns of the optic lobe of *Drosophila* pupae. In this paper we present a real-time image processing method for an ROI to detect cell contour-shapes, using the analysis information, we get and heuristic rules to detect the cell shapes.

Additionally, we have used the signal to noise ratio of the ROI. We conclude that a ($SNR > 4.7$) guarantee a 75% (with a Kappa statistic of 0.50) of accuracy or more in the shape classification using our decision tree rules. One of the main contributions of these algorithms is that we combine automatic operations, with image analysis results, machine learning techniques, and user parameters in real-time to obtain a more reliable result.

Finally, we present a desktop application using Python and OpenCV 3.0. The results in the GUI are visible in real-time, giving the user enormous flexibility and ability to choose the desired ROI and apply post-processing operations. We believe that this GUI can be useful to create a bigger dataset in

order to improve the shape classification even for other applications.

Acknowledgment

We want to thank to Sistema Nacional de investigadores (SNI) CONACYT and Barcelona Supercomputing Center. Sebastián Salazar-Colores (CVU 477758) would like to thank CONACYT (Consejo Nacional de Ciencia y Tecnología) for the financial support of his PhD studies under Scholarship 285651.

References

1. Bao, Q. & Chatziioannou, A. F. (2010). Estimation of the minimum detectable activity of preclinical PET imaging systems with an analytical method. *Medical physics*, Vol. 37, No. 11, pp. 6070–6083.
2. Cohen, W. W. (1995). Fast effective rule induction. In *Machine Learning Proceedings*. Elsevier, pp. 115–123.

3. **Durand, F. & Dorsey, J. (2002).** Fast bilateral filtering for the display of high-dynamic-range images. *ACM transactions on graphics (TOG)*, Vol. 21, No. 3, pp. 257–266.
4. **Horton, J. C. & Adams, D. L. (2005).** The cortical column: a structure without a function. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, pp. 837–862.
5. **Meyer, H., Egger, R., Guest, J., Foerster, R., Reissl, S., & Oberlaender, M. (2013).** Cellular organization of cortical barrel columns is whisker-specific. *Proceedings of the National Academy of Sciences*, Vol. 110, No. 47, pp. 19113–19118.
6. **Morante, J., Erclik, T., & Desplan, C. (2011).** Cell migration in drosophila optic lobe neurons is controlled by eyeless. *The Company of Biologists*.
7. **OpenCV (2015).** OpenCV Itseez.
8. **Rice University, Houston, Texas (2015).** Language and brain: Neurocognitive linguistics.
9. **Sun, J., Liu, C., Bai, X., Li, X., Li, J., Zhang, Z., Zhang, Y., Guo, J., & Li, Y. (2016).** Drosophila FIT is a protein-specific satiety hormone essential for feeding control. *Nature*.
10. **Suzuki, S. et al. (1985).** Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, Vol. 30, No. 1, pp. 32–46.
11. **WEKA (2015).** Weka 3: Data mining software in java.
12. **Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016).** *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

*Article received on 20/10/2018; accepted on 10/02/2019.
Corresponding author is Enrique Javier Arriaga Varela.*