Oecologia Montana 2024,
33, 45-47

TECHNIQUES AND STRATEGIES

Digital image processing for erythrocyte morphometry

D. MATIS[1] and J. NOVÁ[2]

[1]*Department of mathematical methods and operations research, Žilina University, Žilina, Slovak Republic*
[2]*Institute of High Mountain Biology, Žilina University, Tatranská Javorina 7, SK-059 56, Slovak Republic; e-mail: dmatis01@gmail.com*

Abstract. This paper presents an approach to erythrocyte morphometry using digital image processing techniques. The aim of the study is to obtain accurate measurements of the dimensions of oval and nucleated avian erythrocytes, including the length, width and perimeter of the cell membrane and nucleus, by automated analysis. The method is species-independent and can be applied to any species. Digital images of blood smears are processed using Fiji software, where the membrane and nuclei are modelled using two ellipses. The ellipse measurements are then collected and further processed to filter out incorrectly detected cells.

Key words: erythrocyte, morphometry, digital image processing, Fiji software

Introduction

Erythrocyte morphometry, or the measurement of the shape and size of red blood cells (RBCs), is a key aspect of haematology research and diagnosis. Accurate RBC morphometry provides valuable information about various blood disorders and can aid in the early detection and monitoring of disease. Digital image processing techniques have revolutionized the field of biology and medicine by enabling automated and objective analysis of the shape and size parameters of red blood cells. These techniques not only reduce the human subjectivity and potential errors associated with manual analysis, but also allow large numbers of blood samples to be processed in a relatively brief period. The method presented is applicable to all species of ellipsoidal and nucleated erythrocytes. The aim of data processing is to extract dimensions of red blood cells - namely their length, width and circumference - for both the cell membrane and the cell nucleus from the blood smear photographs.

Blood smear photography

Blood smears of birds (in this case *Prunella modularis*) taken using the classical smear method were used to create the photographs. They were subsequently stained by panoptic staining (Doubek *et al.* 2003). The blood smears were examined microscopically using a Leica DM 6000B light microscope (Leica Microsystem, GmbH) at 1000× magnification. Areas of each smear were searched where the erythrocytes were least overlapped and undeformed. Selected areas of the smear were photographed using an additional camera device, LAS (Leica Application Suite; ver.4.5.0; Leica Microsystems CMS GmbH, Switzerland). For each individual, 3 to 4 images were taken (Fig. 1), corresponding to an average of 133 erythrocytes. The number of images taken was based on the quality of the smear. The worse the smear, the more images were taken from separate locations.

Methodology of digital processing of erythrocyte photographs

A minimum of 100 erythrocytes were analyzed for each individual. Erythrocyte perimeter, length and width, and erythrocyte nuclei perimeter, length and width were measured. Due to the need to process a large volume of data, the Fiji (ImageJ) software was used for the analysis. Fiji is licensed as open-source software under the GPL v3 license, and it can be freely used, studied, shared and modified according to the needs of users (Schindelin *et al.* 2012).

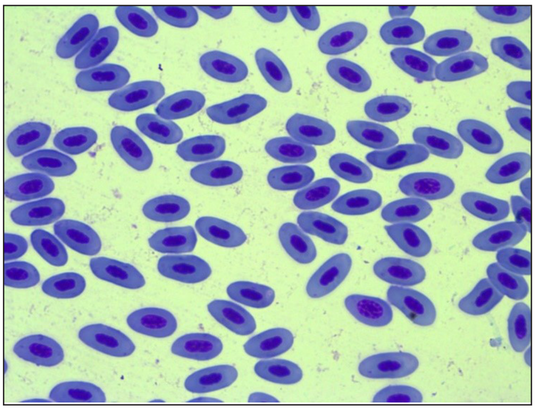

Fig. 1. Example of a photograph of a red blood cell smear with a stained nucleus in the centre taken with LAS.

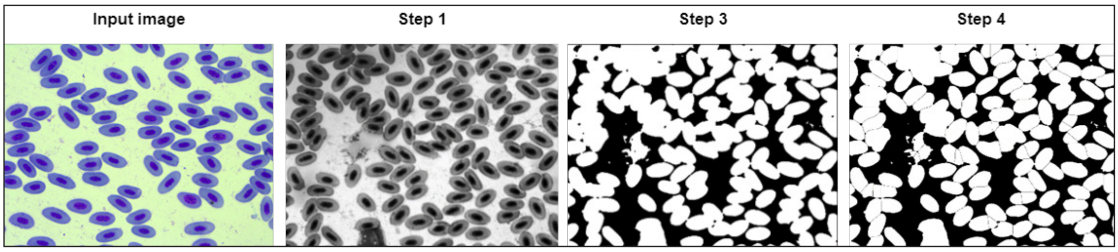

Fig. 2. Visual representation of several steps during the processing of the input image.

This platform provides a robust environment for digital image processing. Users can perform various operations on images, including segmentation, measurement, classification, filtering, morphology, visualization, and more.

The Fiji platform offers various extensions (plugins) that allow for functionality enhancement and customization of the tool according to specific research needs. We utilize an extension called "Biomedgroup," which adds the "Ellipse Split" functionality. Using this feature, we can identify oval shapes in the image.

The process for extracting these dimensions from an image consists of two main steps:

1. *Image Processing:* In the first step, we process the image to identify ellipses present in it. However, since this ellipse detection process does not guarantee accurate cell models, there is a need to filter out some wrong ellipses.
2. *Data Refinement:* The second part of the processing involves cleaning the output data from the first step by removing incorrect ellipses. Subsequently, we pair the ellipses representing the cell membrane and nucleus to create a complete blood cell model.

Image processing

The first step involves processing blood cell images. The processing procedure is as follows:

1. *Basic Image Preprocessing:* To improve performance of subsequent steps, we perform basic adjustments on the image. These include resizing the image (Image > Scale) by half, converting it to grayscale (Image > Type > 8-bit), and enhancing contrast (Process > Enhance Contrast…).
2. *Manual Thresholding:* We manually determine the threshold value using the "Threshold" function (Image > Adjust > Threshold) to create a mask for the blood cell membrane. The output is a binary image.
3. *Median Filtering:* We apply the "Median filter" (Process > Binary > Median) to the mask, which removes noise and smooths edges. Subsequently, we utilize morphological operations such as "Erode" and "Dilate" (Process > Binary > Erode/Dilate) to further reduce noise or fill gaps in the binary mask. We used the median filter with radius of 5 and 4 on the membrane mask and nuclei mask, respectively. The size or number of iterations of these operations depend on the input size of the images.
4. *Watershed Segmentation:* We employ the "Watershed" function (Process > Binary > Watershed) on the membrane mask to separate touching blood cells. In this step, manual intervention using a brush can correct any incorrect divisions (Fig. 2).
5. *Ellipse Detection:* We utilize the "Ellipse Split" function (Plugins > Ellipse Split) from the "Biomedgroup" extension to identify elliptical shapes in the binary mask. The input for this function is the mask obtained from the previous step. For improved results, we can set estimated lower and upper ellipse sizes, significantly reducing their count. In our case, we used a range of 30-350 for the membrane and 25-100 for the nuclei (Fig. 3).
6. *Repeat Steps 2-5 for Nuclei:* For the nuclei of blood cells, we repeat steps 2-5. In step 2, we adjust the threshold boundary to select only nuclei. Step 4 (applying the "Watershed" function) can be skipped during nucleus processing, as nuclei do not touch one another.

The result of the ellipse split operation is a table which contains the position, size, and rotation of all detected ellipses, both for the cell membrane and the nucleus. The content of this table can be saved to a comma separated (CSV) file.

Manually performing the described steps for each image would be time-consuming and prone to user error. Therefore, we have created a macro (Matis 2024), which is a sequence of commands representing our automated steps. The only manual input required from the user is setting threshold values for the membrane and nucleus, as well as correcting errors after the "Watershed" command in step 4. The output is a CSV file containing ellipses for both the cell nucleus and membrane.

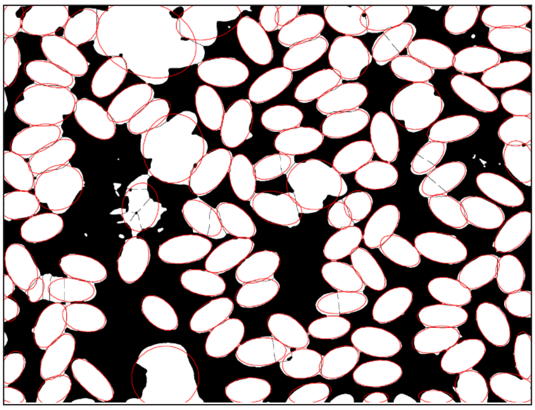

Fig. 3. Final step 5, fitting the ellipses.

Data processing

The output of the image processing consists of ellipses describing both the cell membrane and the nucleus of blood cells. However, at this stage, the membrane ellipse and the nucleus ellipse are not yet connected. Another challenge arises from the possibility of detecting ellipses that overlap entire groups of neighboring cells.

To address these issues, we define several fundamental rules for filtering out incorrect ellipses:

1. *Edge Ellipses*: Ellipses located at the edges of the image often describe only a portion of a red blood cell. Therefore, we remove these edge ellipses.
2. *Overlapping Nuclei*: In certain cases, the nucleus may be composed of two overlapping ellipses, which is incorrect. Therefore, if overlapping nuclei are detected, both ellipses are removed.
3. *Nucleus-Cell Pairing:* We verify the number of nuclei associated with each cell membrane:
   a) Zero Nuclei: If a membrane lacks a nucleus (due to previous steps removing the nucleus), we discard it.
   b) Multiple Nuclei: When one ellipse describes a cluster of closely positioned cells, we encounter the common issue of multiple nuclei. In such cases, we remove the membrane.
4. *Pairing Nuclei and Membranes*: The final step involves connecting each nucleus to its corresponding membrane. The condition is that the entire nucleus ellipse must lie within the membrane ellipse.

The resulting output consists of filtered and paired ellipses that meet the basic criteria for a valid blood cell representation. To implement these rules and pairings, we developed a Python script (Matis 2024). It extends these rules and allows for manual removal of incorrect cells. Importantly, these measurements are in pixels, not physical units. Therefore, when collecting images, attention should be paid to the image scale and proper unit conversion.

Implementing the rules

The foundation for implementing these rules lies in understanding the relationship between two ellipses. Specifically, this involves determining whether an ellipse is fully contained within another, whether it lies outside of another ellipse, or whether the ellipses intersect (Fig. 4).

There are several ways to determine these relationships. In our implementation, we chose a procedure where we transform the boundary of the first ellipse into a finite set of points. Then, we evaluate whether these points lie within the second ellipse (Fig. 5).

Let us consider an ellipse centered at (x_0, y_0) with a rotation angle α, a major axis length *a*, and a minor axis length *b*. To verify whether a point (x, y) lies inside the ellipse, the following condition must hold:

$$\frac{[\cos(\alpha)(x-x_0)+\sin(\alpha)(y-y_0)]^2}{a^2}+\frac{[\sin(\alpha)(x-x_0)-\cos(\alpha)(y-y0)]^2}{b^2}\leq 1$$

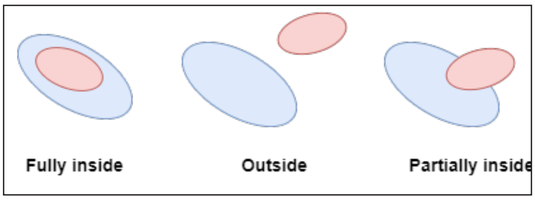

Fig. 4. Three different relationships of two ellipses: ellipse is fully inside / outside, or ellipse is partially inside of another ellipse.

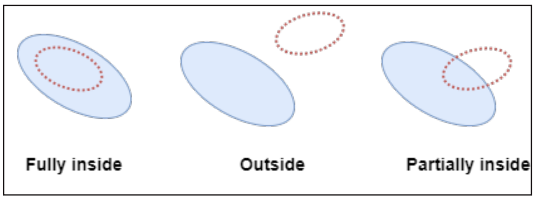

Fig. 5. Approximation of an ellipse by a finite set of boundary points.

If this inequality is satisfied, the point is inside the ellipse. If it is greater than 1, the point is outside the ellipse. When the left-hand side evaluates to exactly 1, the point lies on the boundary of the ellipse. By using this condition, we can verify whether the boundary of an ellipse, described by a set of points, lies within another ellipse. If so, we can assert that the first ellipse is contained within the second. If only a portion of the points lies inside the second ellipse, we can conclude that the ellipses intersect. When no points lie within the ellipse, they are disjointed. This method of verifying the relationship between two ellipses is approximate and its accuracy depends on the number of boundary points. The more points describing an ellipse, the greater the precision, but this also increases computational complexity. In our case, we used 32 points, after determining this number to be sufficient for our needs.

Another crucial implementation detail is to check these mutual relationships only between ellipses where it is meaningful. Verifying the relationship between widely separated ellipses is unnecessary, as it is guaranteed that they do not intersect. Checking all pairs of ellipses would only add to computational overhead.

References

Doubek, J., Bouda, J., Doubek, M., Fürll, M., Knotková, Z., Pejřilová, S., Scheer, P., Svobodová, Z., Vodička, R. 2003: Veterinární hematologie. Noviko, Brno.

Matis, D. 2024: Github – RBC-caliper, Online: https://github.com/Ed-zo/rbc-caliper (retrieved: 23.4.2024)

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D.J., Hartenstein, V., Eliceiri, K., Tomancak, P. and Cardona, A. 2012: Fiji: an open-source platform for biological-image analysis. *Nature methods*, **9**: 676–682.

Received 12 March 2024; accepted 3 April 2024.