1 Preprocesamento de Dados / Data preprocessing

1.1 Algorithm: Normalization of Illumination gradient and brightness

1.1.1 Abstract

Grayscale microscopic images often are inconsistent in distribution of brightness on the Image. Higher concentration of cells, or higher density of the solution on one side of the sample, can result in dark areas, which then hinder the work of morphological algorithms, especially ones based on threshold. For efficient results, many image prepossessing techniques are proposed, including Brightness Independent Contrast Enhancement [1] and LBN [2] for grayscale images, and techniques based on L-Channel Gamma Transform [3] for colored images. Those techniques, although successful in normalization of brightness, can lead to data loss, details reduction, or enhancement of insignificant structures. In many applications, especially in the field of studying optic microscope images, the brightness irregularity is not significant enough to use these tools, as their work reduces too many details. To overcome those issues, an algorithm using mean brightness of the area was implemented, modeled on the Dijkstra's path-finding algorithm.This algorithm successfully manages to normalize the brightness on images with high noise and multiple cellular clusters.

1.1.2 Summary

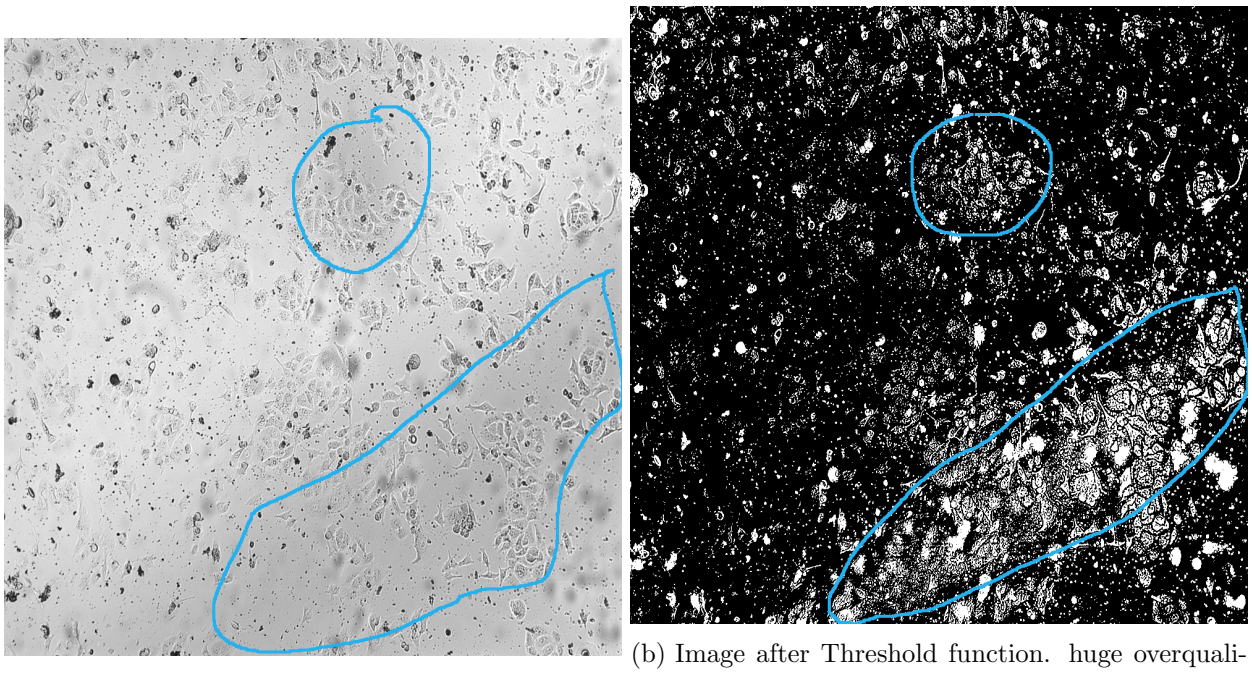
With newer technology and the popularity of AI, the task of normalizing the brightness is often overlooked , as it provides significant value for augmentation techniques[4]. However, in the classical morphological approach, algorithms rely on the consistency of the brightness on the image, especially in operations consisting of noise reduction, segmentation, contrast enhancing[5].

In many of these morphological algorithms, brightness consistency is crucial for accurate work, as they compare each pixel to the same value. On Figure 4, the image with inconsistent brightness was processed using the threshold algorithm. Threshold algorithm separates pixels on image to white or black, by checking if their brightness meets the specific condition (Shown of Figure 3). On Images with brightness instability, the regions which are brighter or darker may suffer from over classification and over qualification by threshold algorithm. This can lead to misinformation in the altered regions

in further analysis, as shown on Figure 4. To compensate this issue, Adaptive Threshold [6] was developed, to calculate the threshold value based on the mean of the surrounding region. This solution can easily operate in regions with unstable brightness, but lacks applicability and is highly independent, as the threshold value is controlled by the algorithm, and can not be set to constant, therefore giving little space to operate on variables.

$$treshold(X, y) = \begin{cases} 1 & \text{for} \quad x \in X; \quad x > y \\ 0 & \text{for} \quad x \in X; \quad x \leq y \end{cases}$$

Figura 1: The variable y is a constant threshold value, to which all pixels/points $x \epsilon X$ are compared. If one side of the image is darker, then the threshold function will overqualify that side of the image [Figure 4]. This might result in over-segmentation in Watershed and over-reduction in noise reduction algorithms.



(a) Original image with inconsistency in brightness.

(b) Image after Threshold function. huge overqualifation problem is evident in the marked region.

Figura 2: The result of brightness inconsistency

In this article, the proposed solution is to smooth the brightness of the image by altering the value of the brightness pf pixels equally within the operated area. This approach does not emphasize any structures, but preserves the initial contrast between foreground and background of the image, altering their brightness simultaneously. This algorithm facilities the work of the threshold algorithm, amplifying it's results and it's credibility. The algorithm divides the image into areas, traversing

through them in a way modeled on Dijkstra's Path Finding algorithm[7], and calculates the mean brightness of surface, altering the mean of studied surface, and striving to the mean brightness of while image in every area.

1.1.3 Algorithm breakdown

The algorithm calculates the mean brightness of image, which is set as goal brightness to all the regions. Then, the area on the image is found, which meets the condition of having the closest mean brightness to goal brightness. The Center point of that surface is a starting point for traversing modeled after Dijkstra's path finding [Figure 7 a]. The algorithm traverses 4 ways, or 8 ways, originating from the mentioned starting point, goes up/down/left/right, in 4-ways version, in 8 ways version also the corners, top-left/top-right/bottom-left/bottom-right, are also included. It collects information about each traverse direction visited points, to not process ones already altered[Figure 7 b, c ,d]. With this approach, each radial area on the image will be altered 4 or 8 times. On each iteration, the algorithm calculates the mean of surface, and, in its initial form, alters each pixels on the area equally [Figure 5].

$$Alter(X, b, r, center) = \begin{cases} x + [b - mean(X, r, center)]; \quad x \in X; & distance(center, x) \leq r; \\ x; \quad x \in X; & distance(center, x) > r; \end{cases}$$

Figura 3: Mathematical representation of a single iteration in the algorithms pure form .Each point (x) of the Image (X) is altered by the difference between goal brightness (b) and mean of Area. If the radius is bigger, then no change in point x is conducted.

The algorithm implemented as shown on Figure 5 is successfully normalizing brightness on the image, but it leaves "leftover circles"on the image [Figure 8]. To avoid creation of these structures, two actions were taken. First, the precision of image was raised from UInt8 to Float. This let the algorithm alter the brightness by fractional values instead of integers. The next implementation, which could solve this issue, is reducing the traverse radius of areas. Although this solves the issue, it is inefficient, as the time complexity of algorithm is $O(n^4)$ for the 4-way version, where $n = width/radius$. As the

result, any changes downgrading the radius are time consuming. Another idea, successfully avoiding influencing time complexity, is radius percentage. Instead of adjusting all points on the area equally, it creates gradient of force, assuring that the algorithm changes points based on their distance from the center of the area. With this approach, the borderline points will undergo little to no change compared to the background, while the mean of the area is still altered towards the global mean [Figure 6].

$$force(x, center, r) = (r - distance(center, x)) \quad /r;$$

(a) force function

$$Alter(X, b, r, center) =$$

$$\begin{cases} x + [b - mean(X, r, center)] * force(x, center, r); & x \in X; \quad distance(center, x) \leq radius; \\ x; \quad x \in X; & distance(center, x) > radius; \end{cases}$$

(b) Changed Alter function

Figura 4: Changed version of Alter function to avoid creation of circular noise on the image

Further changes to the algorithm were implemented, giving a threshold value to the calculation process of the mean of area. With this threshold, points with radical difference of the brightness in comparison to the mean, do not influence the mean value, but they still undergo the change of the brightness. Therefore, the algorithm implemented in this article has 3 variables: traverse, radius, and mean threshold.

The traverse distance and radius of altered surface are variables, which should be defined uniquely for each problem. By fine-tuning, it was discovered that for most microscopic images, the traverse radius of 10 pixels, and altering radius of 15 pixels are the best match. The threshold value was also studied with fine-tuning resulting in difference of 30 being safe for most microscopic images. Therefore, all images present in this article were performed with those values.

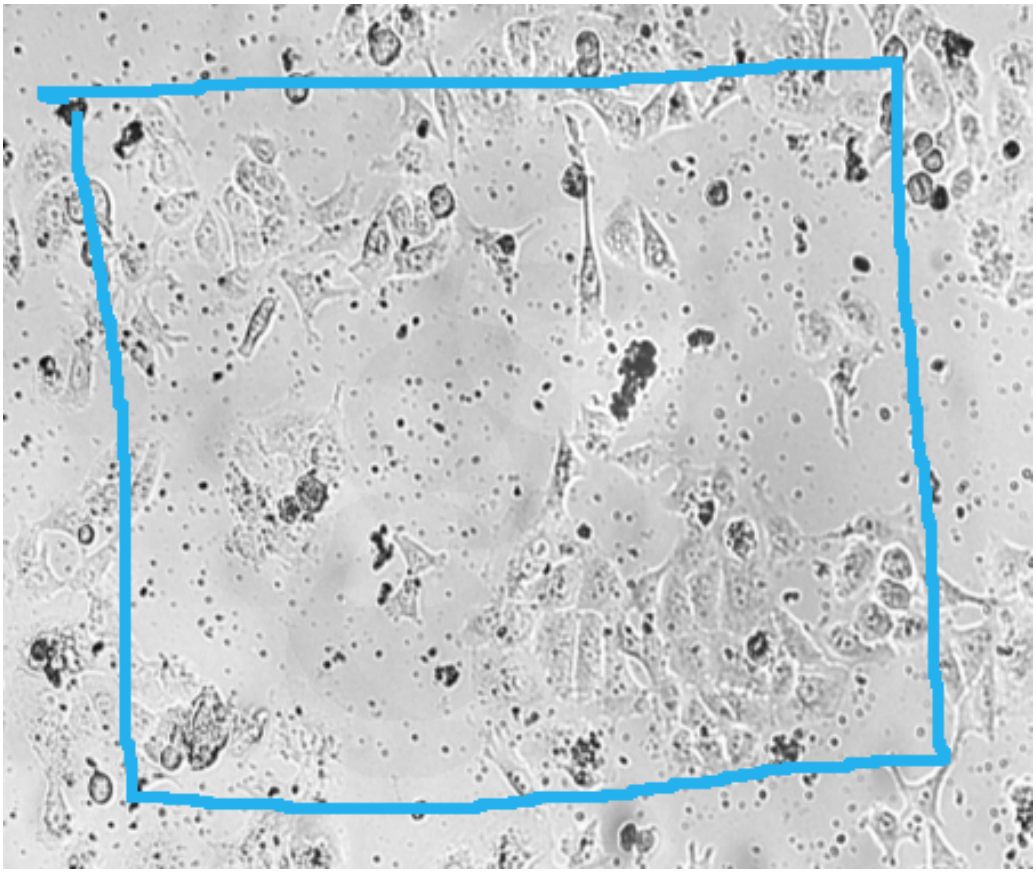
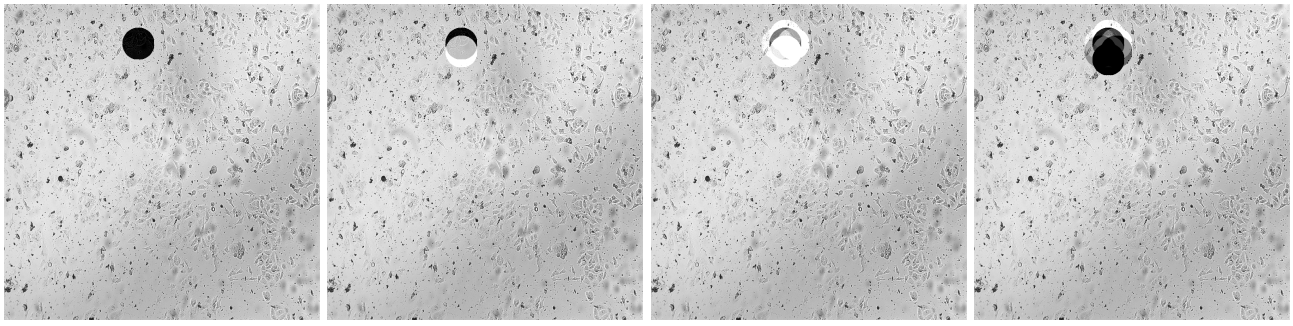


Figura 6: Altered image with visible leftovers



(a) (b) (c) (d)

Figura 5: Visualized work of the algorithm. To make the changes more visible, the brightness was altered by maximum values. In step a), the initial area of set radius (in the example 50 pixels) Is altered. Then the next area is selected in direction bottom, with repositioning of half the radius (25 pixels), with the area rendered staying the same. Then, all the directions from the initial area are altered. Then in d), from the subsequent area the same action occurs. The recursion stops when an area reaches the image boundaries. Once all subsequent areas have stopped, the algorithm completes.

1.1.4 Result:

The algorithm is very efficient in biological images with various changes of brightness on them. In images with a lot of noise, the algorithm works flawlessly [Figure 9 a-c], reducing the instability of

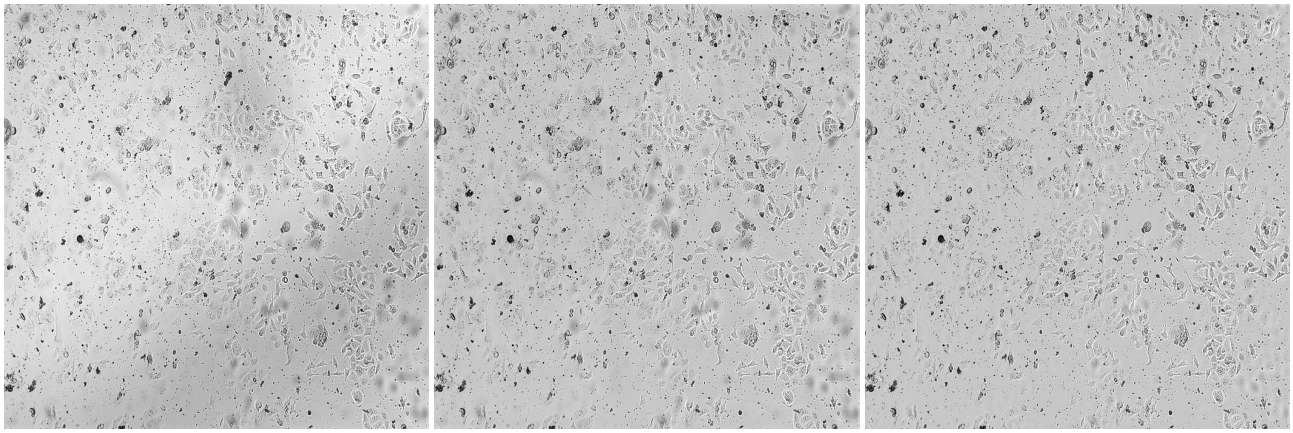
brightness on the image. Applying fine tuning on the altered radius, and traverse radius, can further amplify the results of the algorithm [Figure 9 c]. As we can see in [Figure 9 d-f], for some examples, the parameters of the algorithm after fine tuning can be too extreme, and although the goal of smoothing darker surfaces is reached, the loss of data is visible for unaided eye. Figure 9 g-f shows incapability of algorithm for working with extreme brightness differences, and big objects.

1.1.5 Discussion:

This algorithm shows high efficiency in optic microscopic images with a lot of noise and clusters of cells [Figure 9 a-f]. Implementations, as shown in this article, still leave space for further enhancement of its efficiency and performance. The algorithm is easy to implement and quite rapid, as it is able to process the image of size 1000x1000 in 7,32s, using Intel I3 3 GHz, Integrated Graphic.

Other applications than Images from optic microscope were not studied here, and author does not recommend implementing it in cases where the emphasize of contrast is important, as it can blur the image, resulting in loss of data. The algorithm also does not handle big objects with significant difference in brightness well, resulting in white bubbles, or circular patterns on background [Figure 9 g-i].

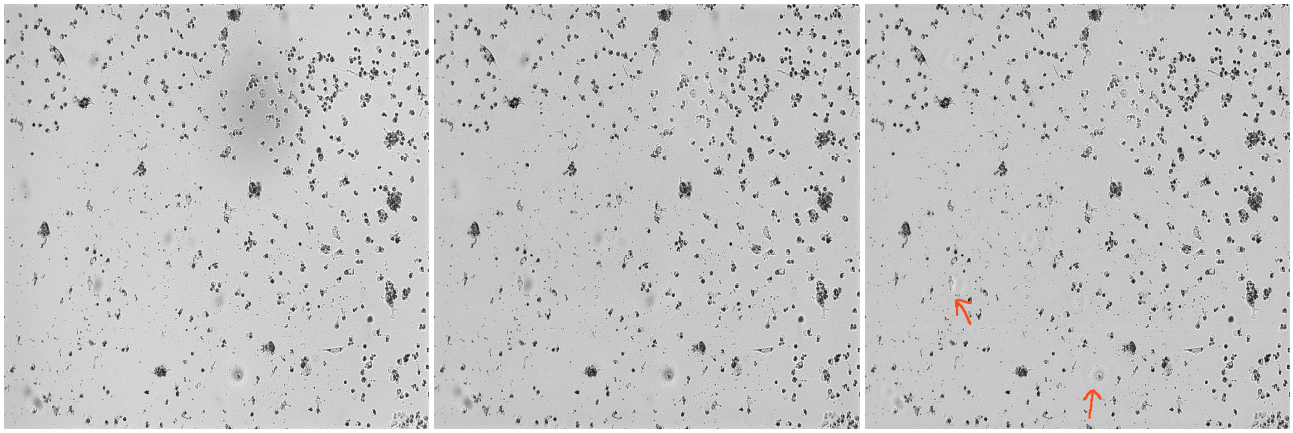
In this article, the algorithm was implemented using basic tools provided by the library openCV, and all the images were generated using C++ on Linux Fedora with a g++ compiler. The test images were shared by Jagiellonian University in Kraków, Poland.



(a) Original Image

(b) Image after work with the algorithm with default parameters of traverse=25, alter=50

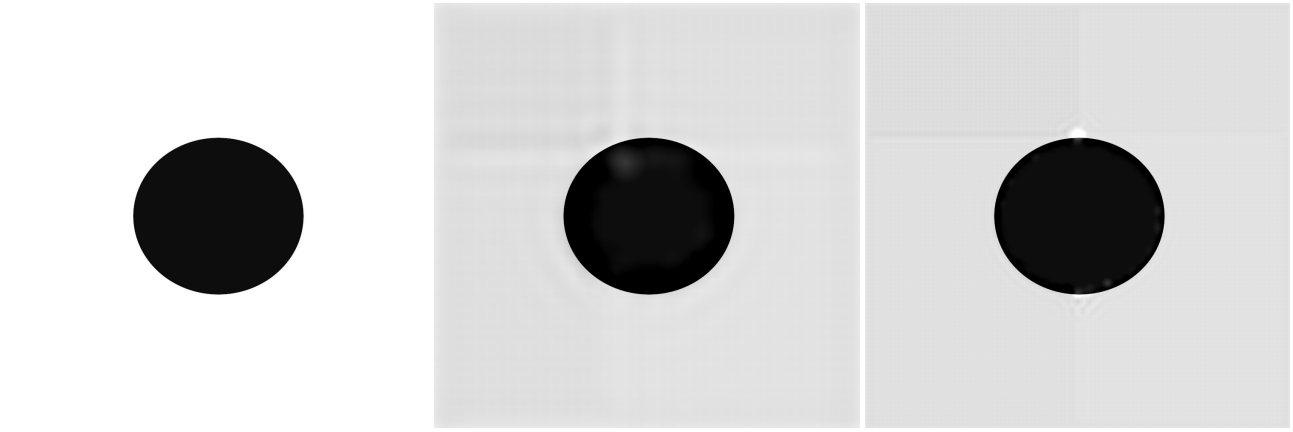
(c) Image after fine tuning, where parameters were changed to traverse=10, alter=15



(d) Original Image

(e) Image after work of algorithm with default parameters of traverse=25, alter=50

(f) Image after fine tuning, where parameters were changed to traverse=10, alter=15. A loss of data is visible



(g) Original Image

(h) Image after work with the algorithm with default parameters of traverse=25, alter=50

(i) Image after fine tuning, where parameters were changed to traverse=10, alter=15.

Figura 7: Images after work of algorithm

1.1.6 Bibliografia

1. F.P .Ph. DE VRIES "AUTOMATIC, ADAPTIVE, BRIGHTNESS INDEPENDENT CONTRAST ENHANCEMENT"Signal Processing Group, Pln'sicc and Acoustic Division, FEL-TNO Physics and Electronics Lahorarorr, Dude Woo lsdorperweg 63, 2509JG The Hague, The Netherlands

2. Mahendra K. Pal and Alok Porwal "A Local Brightness Normalization (LBN) algorithm for destriping Hyperion images"International Journal of Remote Sensing, 2015 Vol. 36, No. 10, 2674–2696, http://dx.doi.org/10.1080/01431161.2015.1043761

3. Min Qi, Shanshan Cui, Xin Chang, Yuelei Xu, Hongying Meng, Yi Wang, Ting Yin "Multi-region Nonuniform Brightness Correction Algorithm Based on L-Channel Gamma Transform", Computational Technologies for Malicious Traffic Identification in IoT Networks https://doi.org/10.1155/2022/2675950

4. Navarun Das, Elima Hussain, Lipi B. Mahanta, "Automated classification of cells into multiple classes in epithelial tissue of oral squamous cell carcinoma using transfer learning and convolutional neural network"Neural Networks, Volume 128, 2020, Pages 47-60, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2020.05.003.

5. S. Beucher, F. Meyer, "The Morphological Approach to Segmentation: The Watershed Transformation", Mathematical Morphology in Image Processing, 1st Edition, 1992, pages 433-481, ISBN: 9781315214610

6. Justin Varghese, Adaptive threshold based frequency domain filter for periodic noise reduction, AEU - International Journal of Electronics and Communications, Volume 70, Issue 12, 2016, Pages 1692-1701, ISSN 1434-8411, https://doi.org/10.1016/j.aeue.2016.10.008.

7. Silvester Dian Handy Permana, Ketut Bayu Yogha Bintoro, Budi Arifitama, Ade Syahputra "Comparative Analysis of Pathfinding Algorithms A *, Dijkstra, and BFS on Maze Runner Game"International Journal Of Information System & Technology Vol. 1, No. 2, (2018), pp. 1-8