2021-11-24

Using the Hough Transform in the Automatization of Contact Angle Calculation

Lazar Milić, Bojan Petrović, Sanja Kojić, Saima Qureshi, Goran Stojanović

Lazar Milić, Bojan Petrović, Sanja Kojić, Saima Qureshi, and Goran Stojanović. 2021. Using the Hough Transform in the Automatization of Contact Angle Calculation. : 1–4. doi: 978-1-6654-2585-8/21.
https://open.uns.ac.rs/handle/123456789/32428

Using the Hough Transform in the Automatization of Contact Angle Calculation

1st Lazar Milić
Department of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences,
University of Novi Sad
Novi Sad, Serbia
lmilic@uns.ac.rs

2nd Bojan Petrović
Department of Dental Medicine
Faculty of Medicine,
University of Novi Sad
Novi Sad, Serbia
ORCID: 0000-0002-3575-3921

3rd Sanja Kojić
Department of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences,
University of Novi Sad
Novi Sad, Serbia
ORCID: 0000-0002-4092-9733

4th Saima Qureshi
Department of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences,
University of Novi Sad
Novi Sad, Serbia
ORCID: 0000-0003-3614-6864

5th Goran Stojanović
Department of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences,
University of Novi Sad
Novi Sad, Serbia
ORCID: 0000-0003-2098-189X

***Abstract*—In this paper an algorithm for automatic contact angle calculation is presented. This algorithm came forth because of the need for a procedure which would minimize the variability that comes with the previously mentioned calculations. With the use of the Hough Transform this has been achieved. In the process of edge extraction, Sobel operators in combination with mean thresholding and morphological operations were implemented. Afterwards, Hough transforms for line and circle were used. With the use of analytic geometry on these features, contact angle was calculated. The described algorithm is compared to the ImageJ plugin for contact angle measurement.**

Keywords— Hough transform, image analysis, contact angle, wetting, goniometry, algorithm

I. Introduction

The Hough transform, an object property extraction method based on a voting technique, has been used for decades in digital image processing as well as a basis in computer vision and image analysis [1]. In recent times, the Hough transform has been implemented in conjunction with neural networks [2] and deep learning [3], opening new horizons for feature detection.

The transform utilizes the fact that a line in the X-Y domain is represented as a point in the parameter domain. The reverse applies as well. Having that in mind, wanted features can be extracted employing geometrical relations which represent previously mentioned features. The parameter domain is divided into a grid. When a line or a curve passes through a cell in the constructed grid, the value of the cell is incremented. The value of a certain cell increases as the number of lines or curves pass through it. After the previously explained voting process is over, a specified number of cells with the highest vote count are chosen. The amount of chosen cells solely depends on the feature that ought to be extracted. Subsequently, relations which connect the X-Y domain with the parameter domain are applied with the goal of acquiring the right features [4].

Wetting is the process of fluid interacting with a surface of the materials, and it is very important in numerous practical applications and in many biomedical and engineering systems. The contact angle is the angle at the interface where water, air and solid meet, and its value is a measure of how likely the surface is to be wetted by the water [5]. Due to its simplicity, the most widely used method for surface-wetting description is sessile-drop goniometry. The above mentioned method calculates the contact angle from the shape of the droplet and can be applied to all sorts of materials including biological surfaces to more complex composites, polymers, metals, crystals, ceramics and minerals. As already observed this apparent simplicity of the method is to some extent misleading, and obtaining meaningful results requires minimization of random and systematic errors [5]. The protocols for performing reliable and reproducible goniometry measurements have been established. These protocols are designed for water as a wetting liquid, however in many scientific disciplines and for many other liquids of different physical, chemical and rheological properties it is necessary to determine the contact angle with different structures. The criteria from the mentioned protocols related to water must not be unreservedly accepted, but the testing of each new liquid requires calibration, standardization and possible corrections of some parameters of the experimental protocol. Since the technique relies on image processing, it has some inherent errors primarily due to optical limitations, particularly in the proximity of the three-phase contact line [6]. Traditionally, the determination of the contact angle is based on the analysis of the image or film obtained during the experiment. It is common for users to enter parameters themselves, which can be subjective and biased. There is a need to reduce this error that can be influenced on, so the development of algorithms that would automate the whole process is highly desirable.

In this paper, the use of the Hough transform on images of droplets on different surfaces will be explored, with the goal of detecting lines and circles, moreover features, leading to the computation of left and right contact angles between the aforementioned surface and droplet. Similar work has been done before in [7]. The algorithm has been developed using Matlab software. Before the implementation a sample of

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 951747.

978-1-6654-2585-8/21/$31.00 ©2021 IEEE

images were taken for testing and measurement. Following the implementation, the output values were compared to the ImageJ software, and their speed, user interaction and variability were compared.

II. Theoretical Basis

The main technique, briefly described in the introduction, the Hough Transform, in its generalized form is used to extract any kind of curves, features from an observed image that do not necessarily have an analytic description. It is based on the template matching technique, where 3 main parameters are defined for any shape $\{y, s, \theta\}$. The priorly mentioned parameters are described as follows: the first one, y determines the reference point, origin; parameter s is the set of scale factors, and Θ is the orientation of the designated shape. Aside from the edge image, the reference table (R-table), which fully defines the origin point y, is needed for the accumulator matrix. The accumulator matrix cells are then incremented if the corresponding edge and parameters align. In the following paragraph the classical Hough Transform pertaining to a circle and a line, will be analysed.

The basis of the Hough Transform for line detection is the fact that one and only one line passes through two points in a plane, on the other hand infinitely many lines pass through one point in the same plane. Using this, and the fact that each line is defined by its intercept n, and its slope, k, as seen in (1), the observed line can be translated into k-n parameter space from its respective x-y space. Analogously, the observed line, which contains infinitely many points, when translated from the x-y domain into the parameter k-n domain will be transformed into a point, with coordinates (k, n). This observed point (k, n) will have infinitely many lines intersecting in it. If this domain is discretized, divided into cells, each time a line, which in the x-y domain represents a point, passes through one of the cells, its value will increase. It is important to note that each cell represents a point in its analogous continuous domain with coordinates (k, n).

$$y = kx + n \tag{1}$$

When calculating the contact angle, it is preferred for the line depicting the border between the surface and the other domains to be parallel to the image border. If the observed line is parallel with the ordinate, its k parameter approaches infinity, which is a roadblock when voting in the parameter domain. This restriction produces the need to translate the line from the x-y domain into the polar coordinate (ρ,θ) system, using a different relation, as seen in (2). This bypasses the previously mentioned problem, by transforming points from the x-y space into sine waves in the ρ-θ space, instead of transforming them into lines in the k-n space. Afterwards the voting process is done in the same way. Use of the ρ-θ space has its drawback, more precisely, it demands a complex inverse calculation to return the parameters of the line back to the x-y space, after the voting process is done.

$$x\cos(\theta) + y\sin(\theta) = \rho \tag{2}$$

When extracting a circular-like feature from an edge image, using the Hough Transform, three parameters must be taken into consideration. Translating a line from the x-y domain into the parameter domain, needs two parameters, k and n, on the other hand, when translating a circle from the x-y domain into the parameter domain, three parameters are needed, radius R, and the circle centre coordinates q and p. This effectively transforms a 2 dimensional space defined with x and y, into a 3 dimensional space defined with q, p, R. The used relations for this process are seen in (3), (4), (5) and (6). This 3-dimensional space is discretized and each cell in it represents a potential circle centre of a defined radius in the 2-dimensional domain. Cell value increases by 1, each time a circle passes through it. When all circles are drawn for a specified radius, the value of the radius is incremented and the process continues, until the radius reaches a previously defined upper limit. After the voting process is finished, the cells are sorted, and the cell with the maximum value in it is chosen. The coordinates for the circle centre are extracted, as well as the radius R.

$$(x-p)^2 + (y-q) = R^2 \tag{3}$$

$$x = p + R\cos(\theta) \tag{4}$$

$$y = q + R\sin(\theta) \tag{5}$$

$$\theta \in [0,360] \tag{6}$$

Having extracted both the line and the circular section from the edge image, its points of intersection are calculated. Circle tangents in the calculated points are determined. By using the information related to the tangent slope k, and the fact that the angle between two lines, where one is parallel to the image border, is equal to the inverse tangent of the slope of the other line, the contact angle is determined.

III. Algorithm Concept

The contact angle calculation algorithm consists of 4 main sections integrated into a GUI using Matlab, illustrated in Fig. 1.

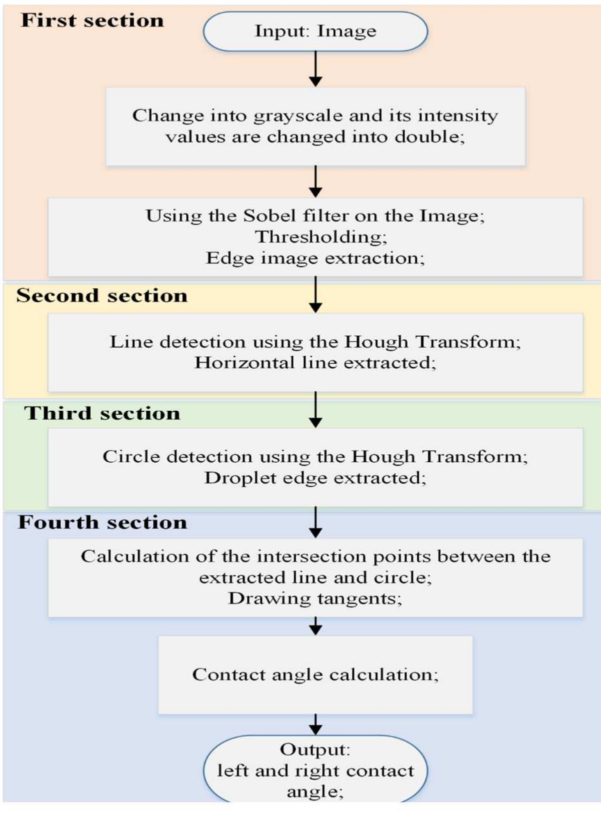

Fig. 1. *Main algorithm*

These sections are:

1. Image preparation and edge extraction;
2. Hough Transform for line detection;
3. Circular Hough Transform;
4. Contact Angle calculation.

The first section, image preparation and edge extraction, consists of five steps:

1. Loading the image;
2. Converting into grayscale and its values into double;
3. Forming the Sobel filter and calculating the first derivative of the image;
4. Defining a threshold;
5. Thinning the resulting edges by using morphological transformations.

At the end of the first section, the result is a binary edge image. This result is fed into the second section with its result being the border line between the surface and the droplet, Fig. 2.

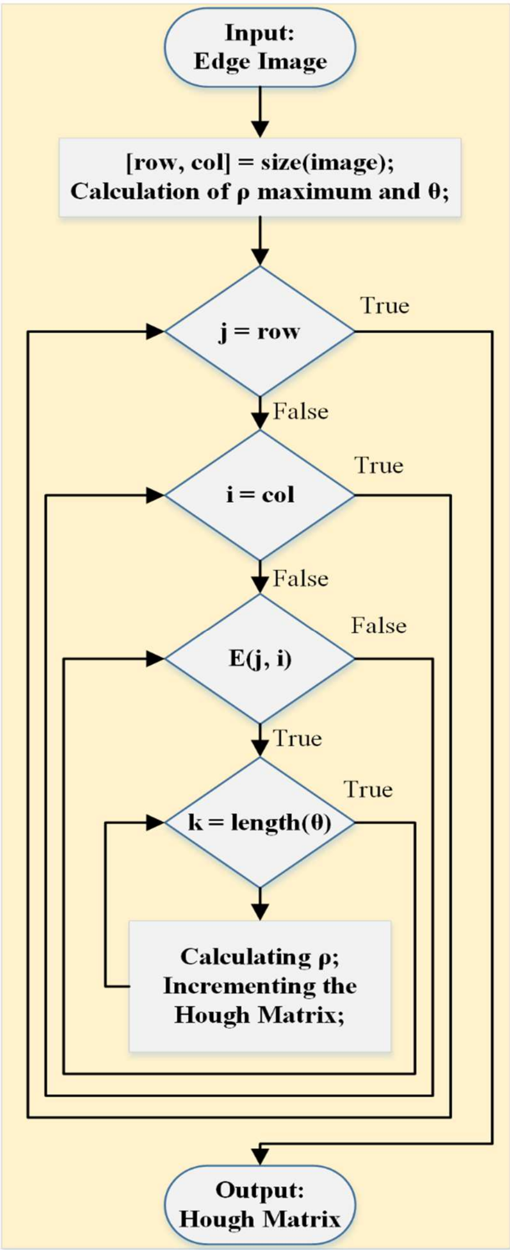

Fig. 2. *The second section detail algorithm*

This section consists of three steps:

1. Calling the Hough Transform function with the binary edge image as the input. This function executes the previously described theoretical basis for line extraction. The output of the function is the Hough matrix, which cell with the maximum value is taken. Its row and column index is taken as well, which represent the corresponding ρ and θ values, that must be transformed into x and y coordinates;
2. Transformation of the ρ and θ values into x and y coordinates, after which the extracted line is formed using its analytical relation;
3. Drawing the line on the input image for visual purposes.

The third section (Fig. 3.) uses the output of the first section as well, and the whole process is visualised on Fig. 4. This time the goal is to extract the edges of the droplet, approximated with a circular shape.

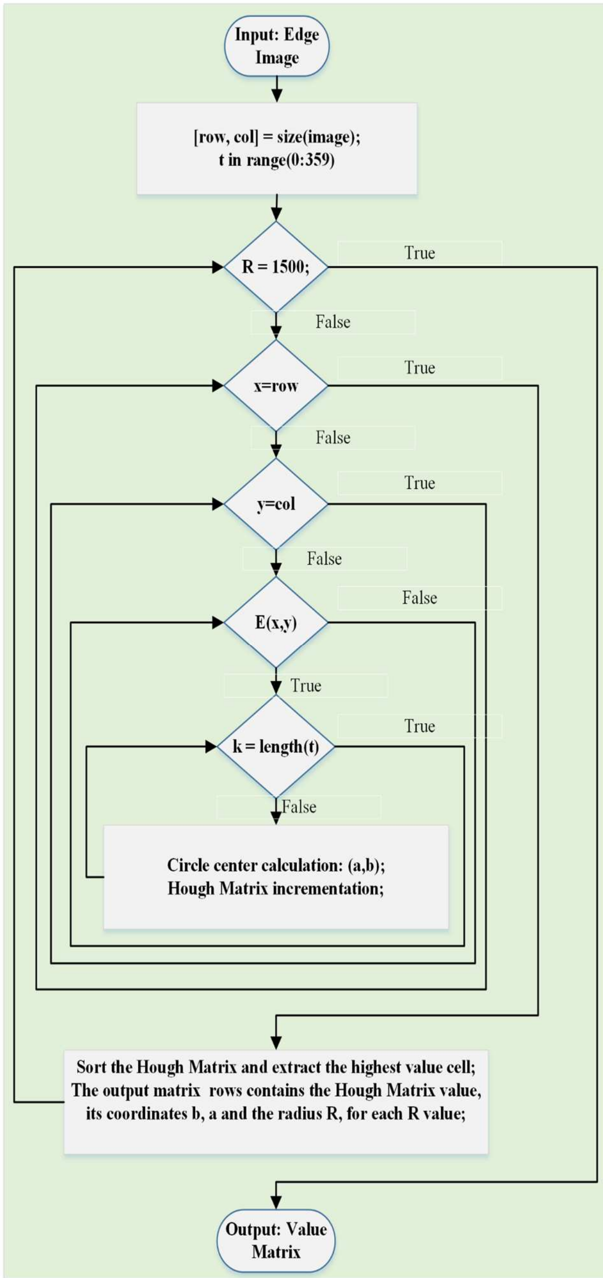

Fig. 3. *The third section detail algorithm*

This section consists of two steps:

1. Calling the Circular Hough Transform with the binary image as the input. After executing the process described in the previous section, Theoretical basis, the output is a box matrix, which contains all the highest valued cells for a given radius, over a range of radii. Aside from the radius and cell value, the aforementioned box matrix contains the coordinates of the potential circle centre;
2. From the box matrix, all cell values are compared and the one with the highest value is chosen, its corresponding radius and centre coordinates are taken. The circular curve is then drawn onto the original image using the previously mentioned parameters.

The final, fourth section consists of three steps:

1. Plotting the tangents of the circle at the intersection points, on the original image;
2. Calculating the angles using simple geometric relations;
3. Writing the determined values into a text file.

When comparing results between the proposed algorithm and the ImageJ plugin, the difference did not exceed 4^{o}, which is in complete agreement with the findings in reports using different methodologies [6].

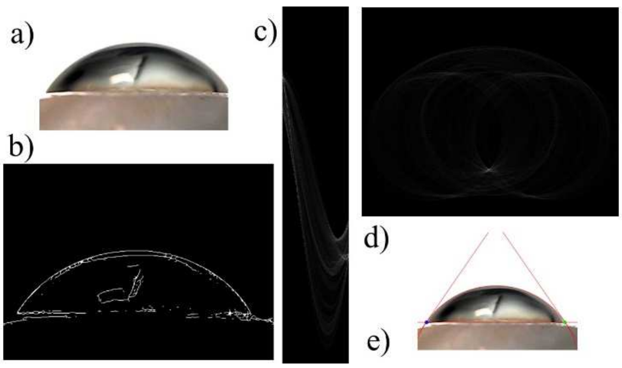

Fig. 4. *a) Input image; b) Edge image; c) Image of the Hough transform for line detection applied on edge image; d) Image of the Hough transform for circle detection applied on edge image; e) Outpun image with angles drawn*

IV. Discussion and Conclusion

Among many open source plugins for contact angle measurement ImageJ is one of the most frequently used in various disciplines including biomedicine, bioengineering, electronics, dentistry, etc. ImageJ is a free image editor widely used for image analysis applications in the biological sciences and beyond. ImageJ attracts user from non-programmers, amateur programmers, and experienced developers equally due to its ease of use, recordable macro language, and flexible plug-in architecture [8]. Allowing such a wide range of users has resulted in a vast community spanning within both biological and physical disciplines. However, a fast rising user base, divergent plugin suites, and technical restrictions have exposed a clear need for a concerted software engineering effort to accommodate developing imaging paradigms and assure that the software can handle the demands of modern science.

Thus, the algorithm described in this paper will be compared to the ImageJ plugin for contact angle measurement. Main motivation for creation of Matlab algorithm for contact angle measurement was to decrease or ideally completely avoid user interaction with the software.

Three parameters are taken into consideration, of which in two, the proposed algorithm has an advantage. This is briefly described in Table I.

TABLE I. Comparison between ImageJ and Matlab

	Matlab algorithm	**ImageJ Contact Angle Plugin**
User Interaction	+	-
Speed	-	+
Variability	+	-

First of all, user interaction in the Matlab algorithm is minimal, reduced to running the software and choosing the image. On the other hand, the contact angle measurement plugin for ImageJ, requires a process of marking the boundaries on the selected image, after which the process is executed. Depending on the approximation method it will use, the user needs to mark the boundary with 7, or with 5 points. This increases the variability of the results, which is the next parameter. Variability is an important factor in measuring dependable results. The final parameter is speed. The Matlab algorithm is much slower, as it takes around 5 minutes to calculate the contact angle, whilst the ImageJ plugin takes on average 10 seconds to complete. The sole hindering factor for this delay is that it takes a long time to fit the best radius during circular feature extraction. This can be modified inside the algorithm by changing the end points of the interval through which the algorithm goes through while it is fitting the radius. The greater the range, the slower the algorithm is. As the main goal of this algorithm was to make a user-friendly way to calculate the contact angle, with as minimal user interaction as possible, the algorithm speed had to be sacrificed. In future iterations of the algorithm, this deficiency will be addressed.

References

[1] R. O. Duda and P. E. Hart, ‘Use of the Hough transformation to detect lines and curves in pictures’, Commun. ACM, vol. 15, no. 1, pp. 11–15, Jan. 1972, doi: 10.1145/361237.361242.

[2] J. Basak, ‘Learning Hough transform: a neural network model’, Neural Comput., vol. 13, no. 3, pp. 651–676, Mar. 2001, doi: 10.1162/089976601300014501.

[3] K. Zhao, Q. Han, C.-B. Zhang, J. Xu, and M.-M. Cheng, ‘Deep Hough Transform for Semantic Line Detection’, IEEE Trans. Pattern Anal. Mach. Intell., vol. PP, May 2021, doi: 10.1109/TPAMI.2021.3077129.

[4] R.C. Gonzalez, and R.E. Woods. Digital Image Processing, 3rd edn. Prentice-Hall, Inc., 2006. ISBN 978-0-13-335672-4

[5] T. Huhtamäki, X. Tian, J. T. Korhonen, and R. H. A. Ras, ‘Surface-wetting characterization using contact-angle measurements’, Nat. Protoc., vol. 13, no. 7, pp. 1521–1538, Jul. 2018, doi: 10.1038/s41596-018-0003-z.

[6] M. Vuckovac, M. Latikka, K. Liu, T. Huhtamäki, and R. H. A. Ras, ‘Uncertainties in contact angle goniometry’, Soft Matter, vol. 15, no. 35, pp. 7089–7096, Sep. 2019, doi: 10.1039/C9SM01221D.

[7] B. Xu, D. Liu, G. Xu, X. Zhang, and L. Bi, ‘A measurement method for contact angle based on Hough Transformation’, Measurement, vol. 46, no. 3, pp. 1109–1114, Apr. 2013, doi: 10.1016/j.measurement.2012.10.023.

[8] C. Rueden, J. Schindelin, M. Hiner, B. DeZonia, A Walter, E. Arena, K. Eliceiri, ‘ImageJ2: ImageJ for the next generation of scientific image data’, BMC Bioinformatics, vol. 18, no. 1, pp. 1–26, Nov. 2017, doi: 10.1186/s12859-017-1934-z.