International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 4, Issue 4, 2017, PP 15-21
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: http://dx.doi.org/10.20431/2349-4859.0404002
www.arcjournals.org

To Measure the Perimeter of an Ellipse Using Image Processing and Mathematical Reasoning

Mohammad Samadi Gharajeh

Young Researchers and Elite Club, Tabriz Branch, Islamic Azad University, Tabriz, Iran

***Corresponding Author:** ***Mohammad Samadi Gharajeh**, *Young Researchers and Elite Club, Tabriz Branch, Islamic Azad University, Tabriz, Iran*

Abstract: *Image processing and mathematical reasoning are two powerful techniques to solve some of the complex problems. Since the perimeter of ellipse is not determined accurately as well as the existing perimeter equations are too complex, researchers attempt to find proper solutions for this issue. This paper proposes a novel approach to measure the perimeter of an ellipse by using image processing and mathematical reasoning. This approach consists of two stages. In the first stage, value of a pixel is calculated via a pixel-by-pixel image processing based on the perimeter of several circles having different radii. In the second stage, the perimeter of an ellipse is defined by the pixels of various ellipses having different diameters and the value of a pixel through a mathematical reasoning. Simulation results show that* $P = 1.14167\pi (a + b)$ *is the suggested perimeter of an ellipse according to the considered simulation scenarios.*

Keywords: *Perimeter of an Ellipse, Image Processing, Mathematical Reasoning, Simulation.*

1. Introduction

A 2D continuous digital image a[x, y] is composed of *m* rows and *n* columns where x = {1, 2, …, m} and y = {1, 2, …, n}. RGB and CMYK are two main color spaces that indicate each pixel of a digital image. RGB pixel is mixed by the red, green, and blue colors. In contrast, CMYK pixel is mixed by the cyan, magenta, yellow, and black colors. Image processing studies any procedure that takes an image as input and returns an image as output. It has various applications in mathematics, biology, biometrics, etc. Some of the mathematical techniques can use a pixel-by-pixel image processing to gather graphical information of the digital images [1-5].

Mathematical reasoning is a powerful tool to evaluate various patterns for selecting some appropriate problem-solving strategies. It can analyze some of the mathematical situations to construct logical arguments. Mathematical reasoning is conducted via four aspects: patterns with explanation, definitions & accurate, covering all cases, and correcting the sequence of results. This type of reasoning uses different thinking skills to carry out the efficient strategies. Thinking skills can be used in thinking processes such as comparing, identifying patterns & relationships, and induction. Pattern with explanation reasoning uses several patterns to conclude a formula for the prediction and validation processes [6, 7].

The perimeter of ellipse is not defined accurately. Therefore, researchers have attempted to find a proper solution for this issue already. In this paper, a new approach is proposed to define the perimeter of an ellipse. It uses value of a pixel based on a pixel-by-pixel image processing through a pattern-based mathematical reasoning. This approach is simulated under various simulation scenarios to define a general formula for the perimeter of an ellipse.

The remainder of the paper is organized as follows. Section 2 represents a problem definition for the perimeter of ellipse. Section 3 describes the proposed approach based on image processing and mathematical reasoning. Section 4 illustrates simulation results under various simulation scenarios. Finally, the paper is concluded by Section 5.

2. Problem Definition

Fig. 1 shows an overall schematic of the ellipse. In this schema, a is the largest diameter and b is the smallest diameter. Since a and b are measured from the center, they are called the radius measures too.

As mentioned before, the perimeter of ellipse is not determined accurately. Furthermore, the existing perimeter equations are too complex. Accordingly, various perimeter equations have been presented by researchers already [8-12]. The proposed approach attempts to find a predictive perimeter of an ellipse based on image processing via a pattern-based mathematical reasoning. The following equation is defined for the suggested perimeter of an ellipse in a way that main objective of the approach is to determine a proper value to x:

$$P = x\pi\,(a + b) \tag{1}$$

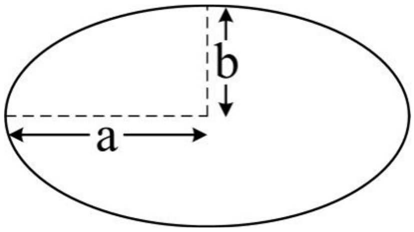

Fig1. *Specifications of an ellipse shape.*

3. The Proposed Approach

This section describes main features of the proposed approach to measure the perimeter of an ellipse. This approach uses image processing and mathematical reasoning to conduct the measurement process. The reasoning process uses patterns with explanation in the perimeter equation based on various patterns. The main underlying idea of this work is that multiplying the number of pixels by value of a pixel can yield the perimeter of any shape. According to this idea, the perimeter of a shape can be given by

$$P = N_p \times V_P \tag{2}$$

Where N_p indicates number of pixels and V_p indicates value of a pixel. The proposed approach consists of two stages. In the first stage, value of a pixel is determined by a pixel-by-pixel image processing according to the perimeter of several circles having different radii. In the second stage, the perimeter equation is defined based on the graphical pixels of several ellipses having different diameters and value of a pixel through a mathematical reasoning process. Fig. 2 shows graphical representation of the above stages.

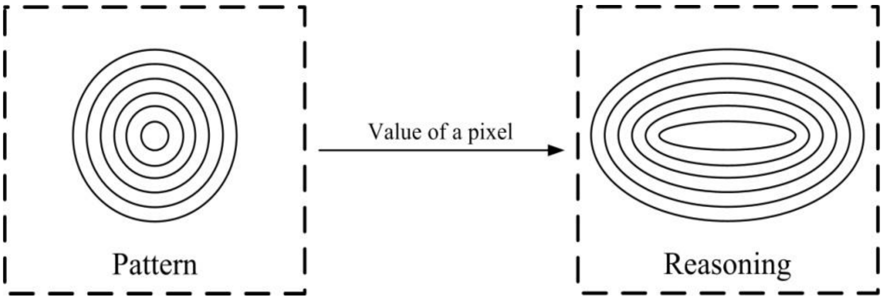

Fig2. *An overall view of the proposed approach*

In the first stage, several patterns of the circles having different radii are used to calculate value of a pixel. This procedure is conducted based on the pixel-by-pixel image processing. Value of a pixel is, in fact, average value of the pixels forming various patterns. Since the perimeter of circle is determined accurately, various patterns of the circle are applied in this approach. Table 1 calculates value of a pixel in the first stage based on area, number of pixels, and average value of a pixel for each circle pattern. Area of a circle is given by

$$A_n = 2\pi r_n \tag{3}$$

Where n is number of patterns, r_n is the radius of circle in the pattern n, and π is the ratio of a circle's circumference to its diameter which is usually approximated as 3.14159. Number of pixels is counted via the pixel-by-pixel image processing according to pixels forming the circle patterns. Average value of a pixel is determined for each pattern as follow

$$V_n = 2\pi r_n / p_n \tag{4}$$

Where n indicates number of patterns, r_n indicates the radius of circle in the pattern n, and p_n indicates number of pixels in the pattern n. Finally, value of a pixel is calculated based on the average values of all patterns as

$$V = (V_1 + V_2 + \dots + V_n) / n \quad (5)$$

Where n is number of patterns and V_n is average value of a pixel in the pattern n.

Table1. *Elements of the first stage specified for the proposed approach*

r	Area	Number of pixels	Average value of a pixel
r_1	$2\pi r_1$	p_1	$2\pi r_1 / p_1$
r_2	$2\pi r_2$	p_2	$2\pi r_2 / p_2$
…	…	…	…
r_n	$2\pi r_n$	P_n	$2\pi r_n / p_n$

Algorithm 1 represents how to calculate value of a pixel based on the perimeter of various circle patterns. Lines 1 to 4 define initial values of the variables. Lines 5 to 11 calculate average value of a pixel for each pattern. It is worth to noting that number of pixels forming the various circles is determined based on the pixel-by-pixel image processing. Line 12 calculates the ultimate value of a pixel according to average values of all patterns. Finally, this value is returned to main program by Line 13. Simulation results show that value of a pixel equals 1.12757 where number of patterns is 200.

Algorithm 1. To calculate value of a pixel in the first stage	
1	n ← Number of patterns
2	PI ← 3.14159
3	Sum ← 0
4	i ← 1
5	While (i <= n){
6	r_i ← i
7	p_i ← Number of pixels forming the circle in the pattern i
8	$V_i \leftarrow 2 \times PI \times r_i / p_i$
9	$Sum \leftarrow Sum + V_i$
10	i ← i + 1
11	}
12	V ← Sum / n
13	Return V

In the second stage, the perimeter of an ellipse is calculated based on various instances of the ellipse through mathematical reasoning. This stage determines the value of factor x in the perimeter equation by using value of a pixel and the pixel-by-pixel image processing. Table 2 represents elements of the second stage including a, b, total value of pixels, and average value of factor x to calculate the ultimate value of factor x defined in Eq. (1). Note that a indicates the largest diameter and b indicates the smallest diameter of the ellipse. The values of a and b are determined in a way that the drawn ellipses would not be similar to circle. Total value of pixels is specified for each instance as following

$$t_{a,b} = p_{a,b} \times V \quad (6)$$

Where a indicates the largest diameter of ellipse, b indicates the smallest diameter of ellipse, $p_{a,b}$ indicates number of pixels in the instance (a, b), and V indicates the value of a pixel calculated by Eq. (5). Furthermore, number of pixels is counted by using the pixel-by-pixel image processing similar to the first stage. Average value of factor x is determined for each instance by

$$X_{a,b} = t_{a,b} / \pi(a + b); where\ a = \{3,4,\dots,m\}\ and\ b = \{2,3,\dots,3m/4\} \quad (7)$$

where $t_{a,b}$ is total value of pixels in the instance (a, b), a is the largest diameter of ellipse, b is the smallest diameter of ellipse, and m is a desired number to generate various instances. Finally, the value of factor x is calculated based on average values of all instances as follow

$$x = \left(X_{3,2} + X_{4,2} + X_{4,3} + \cdots + X_{m,3m/4}\right) / \lfloor (3m^2 - 7m + 4)/8 \rfloor \quad (8)$$

Where m represents a desired number to generate various instances of the ellipse, $X_{m,3m/4}$ represents average value of factor x for each instance, and $\lfloor (3m^2 - 7m + 4)/8 \rfloor$ represents the maximum number of instances. It is worth to noting that the maximum number of instances is resulted based on the series n(n+1)/2 as follow

$$n(n+1)/2 = \lfloor(3m/4 - 1)(m - 1)/2\rfloor = \lfloor(3m^2/4 - 7m/4 + 1)/2\rfloor = \lfloor(3m^2 - 7m + 4)/8\rfloor$$

Table2. *Elements of the second stage determined for the proposed approach*

a	b	Total value of pixels	Average value of factor x
3	2	$t_{3,2}$	$t_{3,2} / \pi\ (3 + 2)$
4	2	$t_{4,2}$	$t_{4,2} / \pi\ (4 + 2)$
	3	$t_{4,3}$	$t_{4,3} / \pi\ (4 + 3)$
…	…	…	…
m	2	$t_{m,2}$	$t_{m,2} / \pi\ (m + 2)$
	3	$t_{m,3}$	$t_{m,3} / \pi\ (m + 3)$
	…	…	…
	3m/4	$t_{m,3m/4}$	$t_{m,\ 3m/4} / \pi\ (m + 3m/4)$

Algorithm 2 describes how to calculate value of factor x based on various instances of the ellipse and the value of a pixel. Lines 1 to 6 define initial values of the variables which are used in the algorithm. NI indicates number of instances throughout the procedure. Lines 7 to 18 determines average value of factor x for each instance and sum of all average values. Finally, the value of factor x is calculated in line 19 and, then, is returned to main program in line 20.

Algorithm 2. To calculate value of factor x in the second stage	
1	m ← A desired number
2	PI ← 3.14159
3	V ← Value of a pixel
4	NI ← 0
5	Sum ← 0
6	a ← 3
7	While (a <= m){
8	b ← 2
9	While (b <= (3 × a / 4)){
10	$P_{a,\ b}$ ← Number of pixels forming the ellipse in the instance (a, b)
11	$t_{a,\ b}$ ← $P_{a,\ b}$ × V
12	$X_{a,\ b}$ ← $t_{a,\ b}$ / (PI × (a + b))
13	Sum ← Sum + $X_{a,\ b}$
14	NI ← NI + 1
15	b ← b + 1
16	}
17	a ← a + 1
18	}
19	x ← Sum / NI
20	Return x

Simulation results show that the value of factor x equals 1.14167 based on the assumed simulation scenarios. This value is resulted in the condition that m is 200 and number of instances is 14801. Therefore, the suggested perimeter of an ellipse is given by

$$P = 1.14167\pi\ (a + b) \quad (9)$$

Where a indicates the largest diameter and b indicates the smallest diameter of the ellipse.

4. Simulation Results

Simulation process is carried out in Dev-C++ 4.9.9.2 environment [13] with C++ programming language [14] by using the Graphics library [15]. Fig. 3 illustrates a simulation view of the first stage to determine the value of a pixel where number of patterns is 200. The first column indicates the radius of circle, the second column indicates the area, the third column indicates number of pixels, and the last one indicates average value of a pixel for each pattern. As represented in these results, value of a pixel equals 1.12757 according to the considered simulation model. Table 3 represents some of the simulation results which are generated by the simulation process.

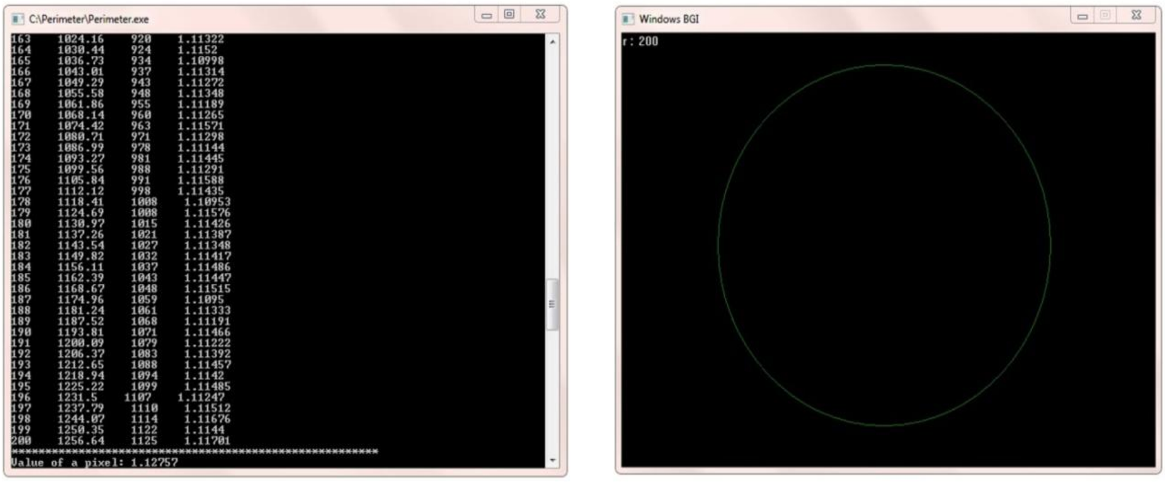

Fig3. *A view of the simulation results to determine the value of a pixel.*

Table3. *Some of the simulation results to calculate the value of a pixel*

r	Area	Number of pixels	Average value of a pixel
1	6.28319	4	1.5708
12	75.3982	64	1.7181
35	219.911	197	1.1163
50	314.159	281	1.118
71	446.106	401	1.11248
80	502.655	450	1.11701
96	603.186	542	1.11289
115	722.566	650	1.11164
134	841.947	759	1.10928
145	911.062	816	1.1165
155	973.894	876	1.11175
178	1118.41	1008	1.10953
185	1162.39	1043	1.11447
195	1225.22	1099	1.11485
200	1256.64	1125	1.11701

Fig. 4 depicts the value of factor x via an instance of the simulation process in the condition that m is 200 and number of instances is 14801. The first column indicates the largest diameter of the ellipse (a), the second column indicates the smallest diameter of the ellipse (b), the third column indicates total value of the pixels forming various ellipses, and the forth one indicates average value of factor x for each instance. Simulation results show that the value of factor x equals 1.14167. Besides, Table 4 represents some of the simulation results related to the second stage for different ellipses.

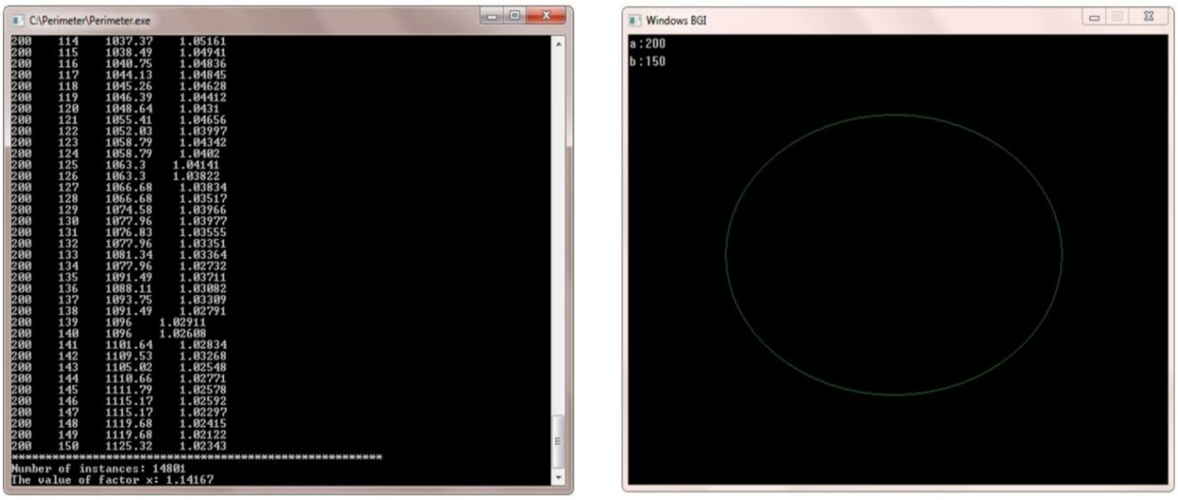

Fig4. *An instance of the simulation process to determine the value of factor x*

Table4. *Some of the simulation results to calculate the value of factor x*

a	b	Total value of pixels	Average value of factor x
3	2	13.5309	0.861402
40	20	200.708	1.06479
52	2	232.28	1.3692
61	17	284.148	1.15958
65	25	312.338	1.10467
70	13	321.358	1.23243
99	21	455.539	1.20836
112	2	504.025	1.40733
119	89	669.778	1.02499
145	60	704.733	1.09426
150	31	690.075	1.21358
160	38	739.688	1.18914
168	80	838.914	1.07675
173	129	970.84	1.02327
180	88	902.058	1.0714
185	138	1040.75	1.02564
190	48	880.634	1.17779
192	87	947.161	1.08061
196	120	1033.98	1.04154
200	150	1125.32	1.02343

5. CONCLUSIONS

This paper proposed a new approach to measure the perimeter of an ellipse through image processing and mathematical reasoning. Since the perimeter of circle is defined accurately, this approach determines the value of a pixel based on various patterns of the circle through a pixel-by-pixel image processing. Simulation results illustrate that the value of a pixel equals 1.12757 where number of patterns is 200. Afterwards, value of the factor x defined for the perimeter equation is calculated based on various instances of the ellipse via the pixel-by-pixel image processing and mathematical reasoning. Simulation results show that the value of factor x equals 1.14167 where number of instances is 14801. Consequently, the suggested perimeter of an ellipse is defined as 1.14167π (a + b) according to the simulation scenarios where a is the largest diameter of the ellipse, b is the smallest diameter of the ellipse, and π is the ratio of a circle's circumference to its diameter.

It is worth to noting that this work does not claim that the suggested perimeter equation is very accurate compared to the existing ellipse equations. In fact, it defines a new perimeter equation by using image processing and mathematical reasoning based on the considered simulation scenarios.

REFERENCES

[1] Annadurai S., Fundamentals of digital image processing, Pearson Education India (2007).

[2] Prashanth H. S., Shashidhara H. L. and Balasubramanya Murthy K. N., Image scaling comparison using universal image quality index, Conference Proceeding, IEEE Int. Conference on Advances in Computing, Control, & Telecommunication Technologies (ACT'09), Trivandrum, Kerala, India, pp 859-863, (2009).

[3] Marion A., Introduction to image processing, Springer, Berlin (2013).

[4] Sonka M., Hlavac V. and Boyle, R., Image processing, analysis, and machine vision, Cengage Learning, Boston (2014).

[5] Burger W. and Burge M. J., Digital image processing: an algorithmic introduction using Java, Springer, Berlin (2016).

[6] Resnick L. B., Education and learning to think, National Academy Press, Washington (1987).

[7] Wong K. Y., Mathematics for science: a Brunei experience, Teaching Mathematics and its Applications. 21(2), 55-65 (2002).

[8] Barnard R. W., Pearce K. and Schovanec L., Inequalities for the perimeter of an ellipse, Journal of mathematical analysis and applications. 260(2), 295-306 (2001).

[9] Villarino M. B., Ramanujan's Perimeter of an Ellipse, arXiv preprint math/0506384, (2008). Available URL: https://arxiv.org/abs/math/0506384v1.

[10] Chandrupatla T. R. and Osler T. J., The perimeter of an ellipse, Math. Sci. 35, 122-131 (2010).

[11] Wang M. -K., Chu Y. -M., Qiu S. -L. and Jiang Y. -P., Bounds for the perimeter of an ellipse, Journal of Approximation Theory. 164(7), 928-937 (2012).

[12] Chung D. D. and Wolfgramm R., Maxillary arch perimeter prediction using Ramanujan's equation for the ellipse, American Journal of Orthodontics and Dentofacial Orthopedics. 147(2), 235-241 (2015).

[13] Laplace C., Berg M., Lai H. and Mandravellos Y., Dev-C++ 4.9.9.2, Free Software Foundation Inc., Cambridge, Massachusetts, USA, (1991). Available URL: http://www.bloodshed.net/.

[14] Alexandrescu A., Modern C++ design: generic programming and design patterns applied, Addison-Wesley, Boston (2001).

[15] Watson M., Portable GUI development with C++, Osborne/McGraw-Hill, New York (1992).

AUTHOR'S BIOGRAPHY

Mohammad Samadi Gharajeh, received ASc in Computer Software from University of Applied Sciences and Technology, Iran in 2005. He received BSc in Software Engineering of Computer from University of Applied Sciences and Technology, Iran in 2009. Afterward, he received MSc in Architectural Engineering of Computer from Islamic Azad University of Tabriz, Iran in 2013. His research interests include control, robotics, embedded systems, artificial intelligence, and learning theory. He is an Editorial Board member and a Reviewer in some of the international scientific journals, a Lecturer of university, and an IEEE Member now.

Citation: *Mohammad Samadi Gharajeh (2017). To Measure the Perimeter of an Ellipse Using Image Processing and Mathematical Reasoning, International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 4(4), pp.15-21, DOI: http://dx.doi.org/10.20431/2349-4859.0404002*