International Journal of Advanced Culture Technology Vol.7 No.3 134-142 (2019)
DOI 10.17703/IJACT.2019.7.3.134

IJACT 19-9-19

Object Recognition using Comparison of External Boundary

Suk Won Yoo

SeoKyeong Univ., Dept. of Software, Seoul, Korea
E-mail: swyoo@skuniv.ac.kr

Abstract

As the 4th industry has been widely distributed, there is a need for a process of real-time image recognition in various fields such as identification of company employees, security maintenance, and development of military weapons. Therefore, in this paper, we will propose an algorithm that effectively recognizes a test object by comparing it with the DB model. The proposed object recognition system first expresses the outline of the test object as a set of vertices with the distances of predefined length or more. Then, the degree of matching of the structures of the two objects is calculated by examining the distances to the outline of the DB model from the vertices constituting the test object. Because the proposed recognition algorithm uses the outline of the object, the recognition process is easy to understand, simple to implement, and a satisfactory recognition result is obtained.

Keywords: *Image Processing, Object Recognition, Accumulation, External Boundary, Edge Detection, Image Analysis, Feature Extraction*

1. Introduction

Today we live in the world of the fourth industry. Almost everything is being executed by automation devices such as artificial intelligence [1], virtual reality [2], internet of things [3], and autonomous navigation [4]. The fourth industry is based on the development of camera equipment and also the development of image processing and image recognition technology with it. Image recognition is a process of recognizing an experimental object captured by camera devices using previously defined DB objects in object recognition system [5]. To do this, the image processing executes some preprocessing steps for the input image in order to make the experimental object to be recognized better [6].

2. The Related Works

Image processing and image recognition [7] are used in many fields today such as industry and broadcasting [8]. In the field of architecture, these technologies are being used to monitor the progress of the construction process [9]. In the medical field, medical image data such as CT, MRI, and ultrasound images are used to examine changes in the patient's health status [10]. In the field of biochemistry, these technologies are used to detect changes in cells or abnormal cells in the process of new drug developments [11]. In addition, drone, a flying object, is used to manage forests and protect coasts [12]. Conventional image techniques normally use the internal skeletal structure of the object to recognize it. In order to get better image recognition result, we propose a new method to recognize an object by comparing it with DB

Manuscript received: July 10, 2019 / revised: July 21, 2019 / Accepted: July 31, 2019
Corresponding Author: swyoo@skuniv.ac.kr
Tel:+82-2-940-7751
SeoKyeong Univ., Dept. of Software, Korea

models stored in object recognition system using only the outline of the object in the experimental image.

3. The Main Subject

The object recognition system extracts outline of the object in the experimental image obtained by the input image device. Then, the vertices forming the outline of the experimental object are selected to create a simplified outline. The object recognition system also defines the shapes of the DB models using a similar method for the DB models. For each vertex constituting the outline of the experimental object, the distance to the line segment passing two neighboring points is calculated. The recognition system calculates the distances and removes vertices with short distances repeatedly until all distances are greater than or equal to a predefined length. In the next step, the object recognition system overlaps the outline of DB model on the vertices forming the outline of the experimental object. Then, for each vertex of the object, calculate the distance to the outline of the DB model. As a result, the average of the distances from the vertices of the experimental object to the outline of the DB model becomes the discordance value of the experimental object and the corresponding DB model.

3.1 Object Recognition using Comparison of External Boundary

The following steps will explain the object recognition algorithm using comparison of external boundary in more detail.

Step 1) The object recognition system detects line segments of an object in the image used as DB model. Then, lines segments representing the external shape of the object are extracted and simplified in order to define the shapes of the object to be used as comparison data. The outline of the object registered as DB model is in the form of 2D black-and-white image, and it is defined by black lines on a white background.

Step 2) The recognition system also detects line segments of the experimental object, and then the system examines the correlation between line segments and creates a simplified outline of the experimental object.

Step 3) The number of vertices composing the external boundary of the experimental object is stored in the variable Number_of_Vertex, and the positions of the vertices are stored in the Distance array. The structure of the Distance array is a record type that stores three pieces of information such as X, Y indicating the positions of vertex in the two-dimensional image, and Distance indicating the distance to a line segment passing through two adjacent vertices.

Step 4) Set the variable Flag_minimum_distance_has_been_found be equal to true.

Step 4.1) While Flag_minimum_distance_has_been_found is equal to true, repeat the execution of the following steps.

Step 4.1.1) For each vertex in the Distance array,

Step 4.1.1.1) Find the previous vertex and the next vertex, and then generate a line segment that pass through these two vertices.

Step 4.1.1.2) Calculate distance from this vertex to the line segment of two adjacent vertices, and then store the distance in the Distance array.

Step 4.1.2) Let the constant variable Distance_Thresholding_Value be the predefined minimum of distance that distances from all vertices to the line segments connecting its two adjacent vertices should be greater than or equal to. Then, find a vertex index number in the Distance array that has the shortest distance.

Step 4.1.2.1) If the distance from the selected vertex to the line segment passing through two adjacent vertices is shorter than Distance_Thresholding_Value, then remove the vertex from the Distance array. Because the external boundary of the object in the test image has been changed by eliminating this vertex, the Distance array should be updated and also distances of all vertices in the array should be recalculated.

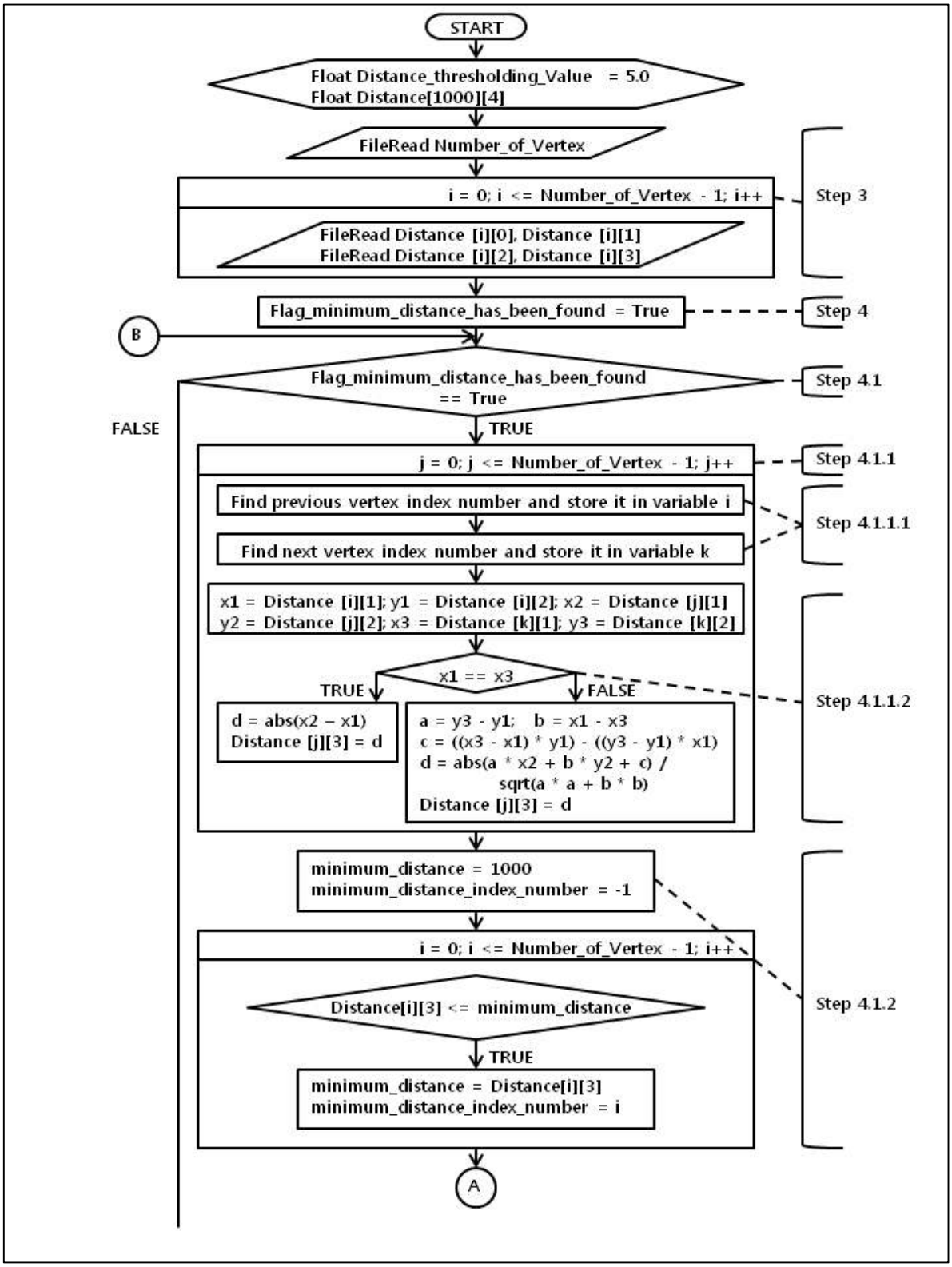


Figure 1. Flowchart

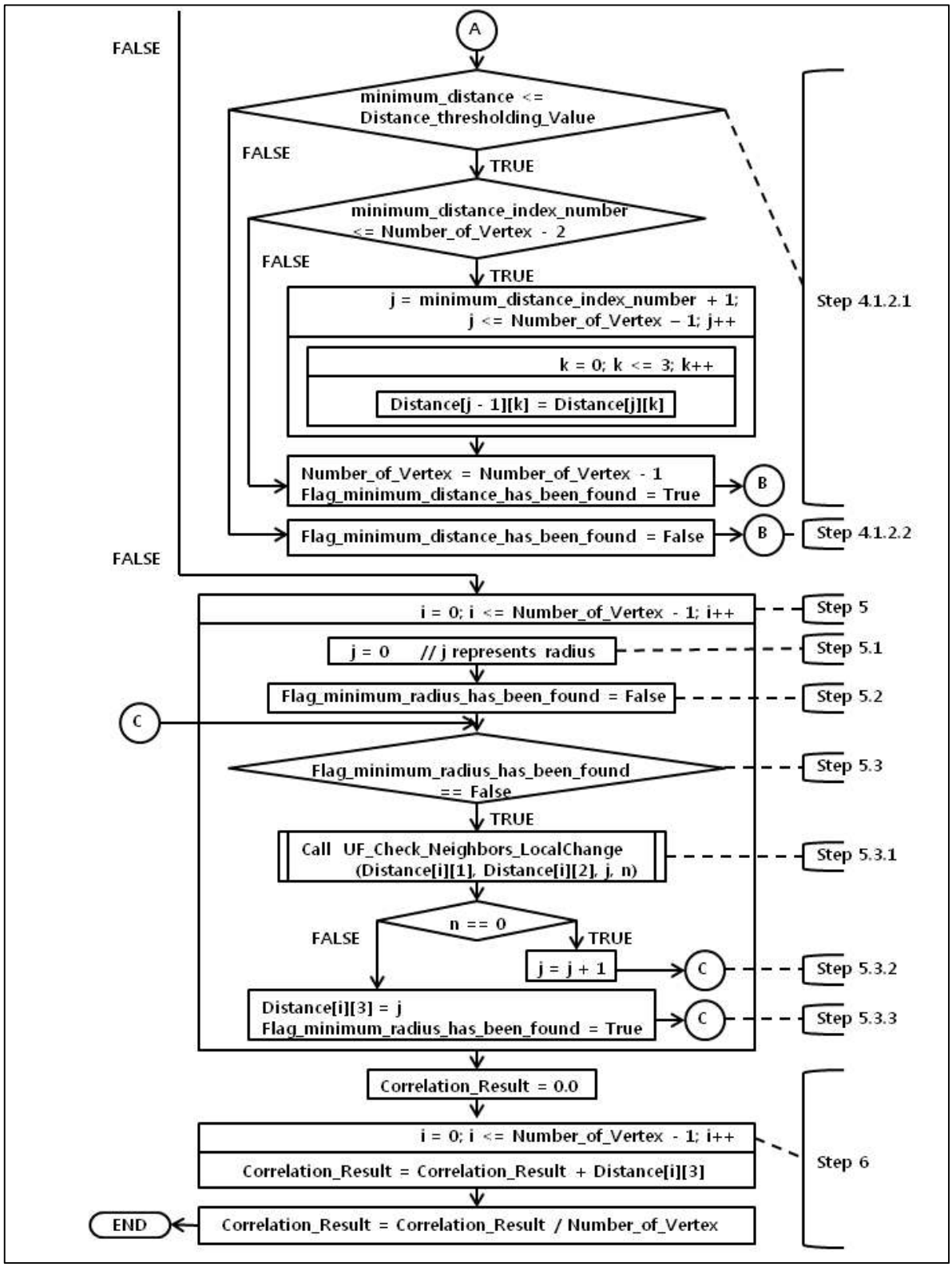


Figure 1. (continued) Flowchart

Step 4.1.2.2) If the distance is greater than or equal to Distance_Thresholding_Value, this case means that all distances from vertices to the line segments connecting its two adjacent vertices are greater than or equal to Distance_Thresholding_Value. Therefore, terminate the repeated execution of these steps and then escape from it by setting Flag_minimum_distance_has_ been_found to be false.
Step 5) For each vertex in the Distance array, find a radius of a circle of the vertex that touches external boundary of DB model by the following steps.
Step 5.1) Let Radius be the radius of a circle of each vertex, and set it to be equal to 0. This Radius will be repeatedly increased by 1 until a circle of the radius touches external boundary of DB model.
Step 5.2) Let the variable Flag_minimum_radius_has_been_found be equal to false.
Step 5.3) While Flag_minimum_radius_has_been_found is equal to false, repeat the execution of the following steps.
Step 5.3.1) Call the function UF_Check_Neighbors_LocalChange(X, Y, Radius, Number_Of_Vertex) to check neighbors of a given vertex, where the parameter X and Y are the position of the vertex in the image, Radius is a radius of a circle of the vertex, and Number_Of_Vertex represents the number of vertex in the intersection of the outline of the circle of the vertex with Radius and the external boundary of DB model.
Step 5.3.2) If Number_Of_Vertex is equal to 0, this means that there is no vertex in the intersection of the outline of the circle of the vertex with Radius and the external boundary of DB model. Therefore, Radius is increased by 1.
Step 5.3.3) If Number_Of_Vertex is greater than 0, this case means that the function UF_Check_Neighbors_ LocalChange() has found one or more vertices on the outline of the circle of the vertex with Radius that touches external boundary of DB model. Therefore, the Radius is stored in the Distance array, and the recognition system terminates the repeated execution of these steps and then escapes from it by setting Flag_minimum_radius_has_been_found to be true. Here, there is one thing to keep in mind that Radius is the distance from the vertex to the external boundary of DB model. Therefore, the bigger the Radius is, the more discordance is.

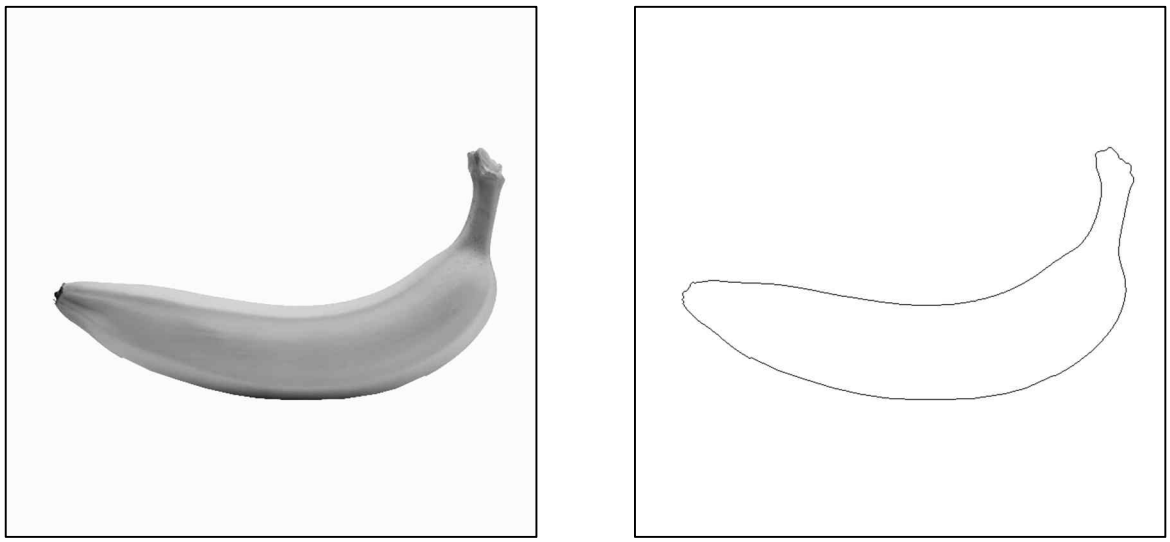
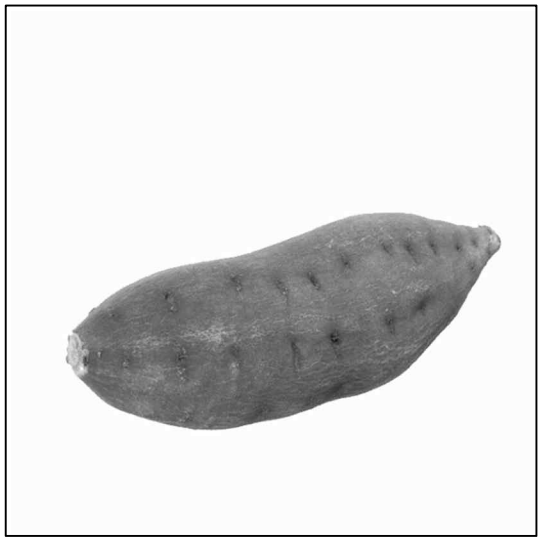


Figure 2. (a) DB Model #1 - Banana (b) Outline of Banana

Step 6) Degree of discordance between DB model and the object in the test image is calculated by checking how much each vertex in the Distance array is far from the external boundary of DB model and measuring the distance from the vertex to the external boundary. In other words, degree of discordance is defined to be the average of the sum of radii of all vertices because radius of each vertex represents the distance from the vertex to the external boundary of DB model. (Flowchart is shown in Figure 1)



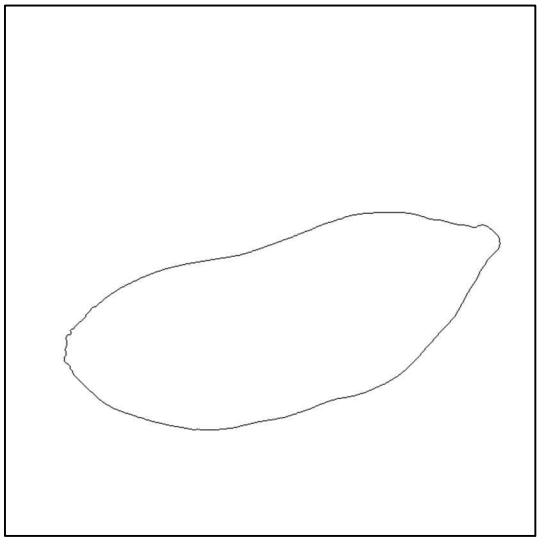
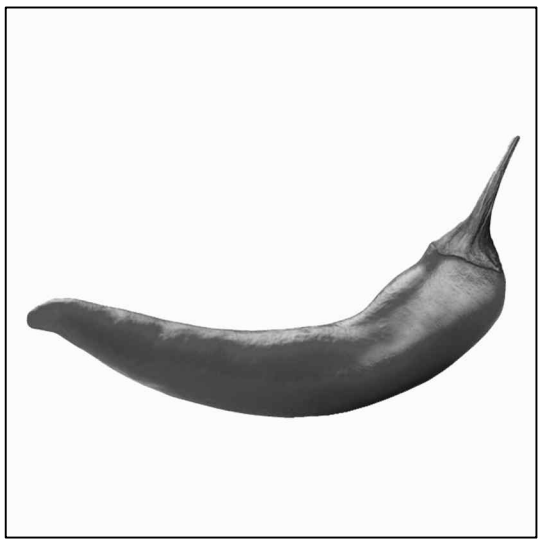


Figure 3. (a) DB Model #2 – Sweet Potato (b) Outline of Sweet Potato

3.2 The Results

The following figures show the results of executing the object recognition algorithm using comparison of External boundary. Figure 2(a) shows the black and white input image of the banana used as DB model #1, and Figure 2(b) does the outline of the banana obtained by applying the Adaptive Thinning Algorithm [13].



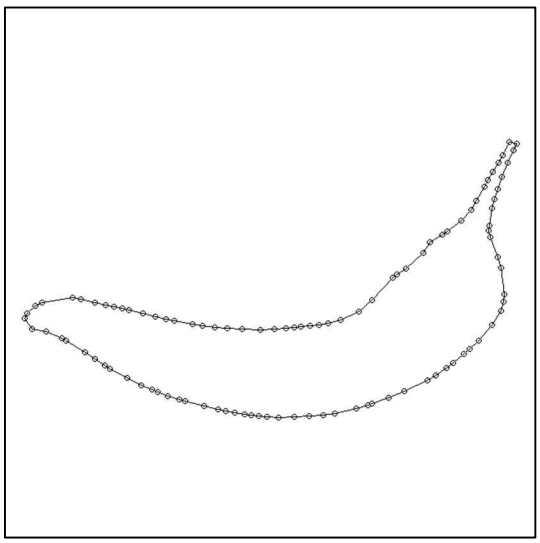


Figure 4. (a) Experimental Image - Chili (b) Outline of Chili with 103 Vertices

Figure 3(a) also shows the black and white input image of the sweet potato used as DB model #2, and Figure 3(b) does the outline of the sweet potato obtained using the same method.

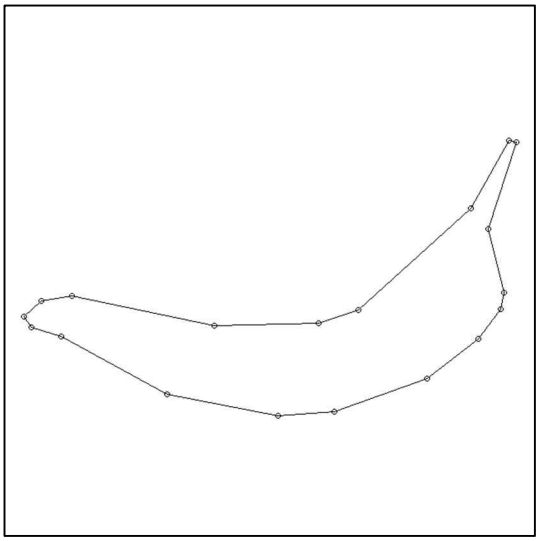


Figure 4. (c) Simplified Outline of Chili with 19 Vertices

Figure 4(a) shows the black and white input image of the chili that the object recognition system should recognize. The recognition system first extracts the outline of the chili by applying Adaptive Thinning Algorithm [8] on the input image. Then, the system examines the patterns of the lines composing the outline of the chili and selects important vertices that make up the outline. Figure 4(b) shows the 103 number of vertices obtained through this process and also the simplified outline of the chili obtained by connecting these vertices in straight lines.

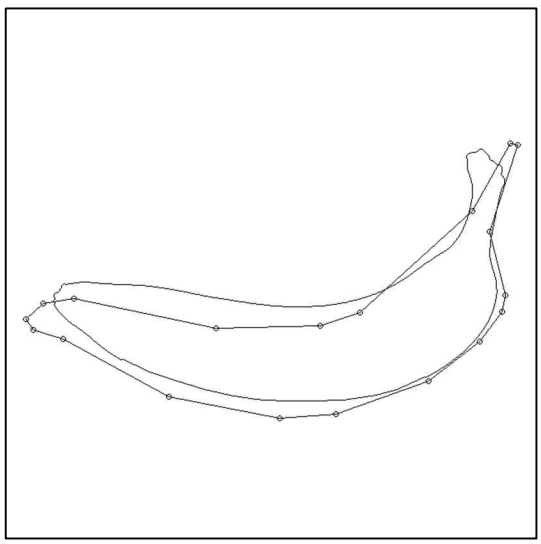
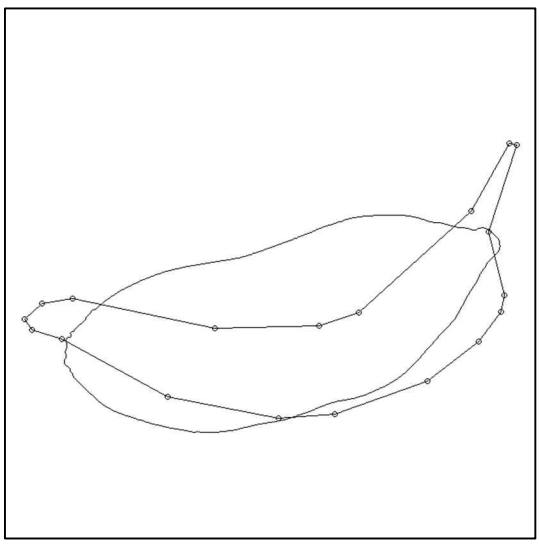


Figure 5. (a) Overlapping of the Outline of DB Model #1 on the Simplified Outline of Chili Composed of 19 Vertices



(b) Overlapping of the Outline of DB Model #2 on the Simplified Outline of Chili Composed of 19 Vertices

For each of the 103 vertices constituting the outline of the chili, the object recognition system checks the distance to the line segment connecting two adjacent vertices and then removes the vertex if the distance is smaller than the predefined constant variable Distance_Thresholding_Value, 5.0. Figure 4(c) shows 19 number of vertices obtained through this process and also the more simplified outline of the chili obtained by connecting these vertices.

The object recognition system recognizes the chili by comparing it with DB models. The recognition

system first overlaps the outline of DB model on the vertices forming the outline of the chili. Then, for each vertex constituting the outline of the chili, draw a circle at each vertex to calculate the distance to the outline of the DB model. This distance is the discordance value of the vertex and the DB model. As a result, the discordance value of the experimental model and the DB model becomes the average of these radii of the vertices forming the outline of the chili. Figure 5(a) shows the overlapping of the outline of the DB model #1 (see Figure 2(b)) on the outline of the chili consisting of 19 vertices (see Figure 4(c)), and Figure 5(b) does the overlapping of the outline of the DB model #2 (see Figure 3(b)) on the outline of the chili (see Figure 4(c)).

Table 1. Distance to DB Models from Each Vertex

Vertex #	X	Y	Distance to DB Model #1	Distance to DB Model #2
0	253	386	36	79
1	81	350	18	27
2	44	356	12	50
3	23	375	38	56
4	32	388	38	43
5	68	399	21	5
6	196	469	21	36
7	330	495	21	3
8	398	490	17	14
9	510	450	5	26
10	572	402	6	40
11	599	366	11	44
12	603	346	10	37
13	584	269	3	1
14	618	164	33	108
15	609	162	29	107
16	563	244	3	21
17	427	367	16	91
18	379	383	24	94

Table 1 shows the location of the 19 number of vertices constituting the outline of the chili and also the distances to the outlines of the DB models at vertex. The chili has a discordance value of 19.052 (= 362/19) with the DB model # 1, banana, and a discordance value of 46.421 (= 882/19) with the DB model # 2, sweet potato. As a result, the target chili is recognized as a banana among these two DB models.

3.3 The Pros and Cons of the proposed Object Recognition Algorithm

Similar to the conventional recognition methods, there are some advantages and disadvantages in the proposed algorithm. Advantages include: 1) It takes less computation time because it uses only the outline of the object. 2) Because it is a comparison method between the outline of the DB model and the vertices composing the outline of the object in the experimental image, it does not matter the order of the vertices defining the outline of the object. The disadvantage is that the shape of the object in the experimental image changes according to the value of the constant variable Distance_Thresholding_Value. Therefore, when the simplified outline of the experimental object is compared with DB models, the discordance value might be calculated differently, and hence it is possible that the experimental object can't be correctly identified.

4. Conclusion and Future work

In this paper, we proposed an object recognition algorithm that ignores the internal structure of an object and uses only a simplified external boundary of the object. The proposed algorithm is easy to understand and generates satisfactory result. The significance of this study is that a test object can be recognized by using the

distance between the vertices representing the outlines of the object and DB model irrespective of the order of the vertices defining the outline of the object. In the future research, we are studying a method of recognizing an object with only partial information in the case that a part of the input image is lost due to a mechanical problem of the input imaging device.

References

[1] D. Rumelhart, J. McClelland, *Parallel Distributed Processing*, MIT Press, pp.121-127, 1987.
[2] A. Wexelblat, *Virtual Reality*, Academic Press, pp.1-73, 2014.
[3] P. Waher, *Learning Internet of Things*, PACKT Publishing, pp.35-64, 2015.
[4] D. Watzenig, M. Horn, *Automated Driving*, Springer, pp. 71-140, 2016.
[5] M. Kang, "Object Recognition Using the Edge Orientation Histogram and Improved Multi-Layer Neural Network", *International Journal of Advanced Culture Technology (IJACT)*, Vol.6, No.3, pp.142-150, 2018.
[6] B. Jeong, M. Kang, Y. Jung, "A Study on the Facial Expression Recognition using Deep Learning Technique", *International Journal of Advanced Culture Technology (IJACT)*, Vol.6, No.1, pp.60-67, 2018.
[7] B. Jahne, *Digital Image Processing*, Springer, pp. 100-133, 2013.
[8] K. Jang, S. Lee, "Analysis of Digital Video with a Focus on the Documentary Colored Hearing", *Journal of the Convergence on Culture Technology(JCCT)*, Vol.2, No.2, pp.23-27, 2016.
[9] W. Pratt, *Introduction to Digital Image Processing*, CRC Press, pp. 139-154, 2013.
[10] C. Leondes, *Image Processing and Pattern Recognition*, Academic Press, pp.22-35, 1998.
[11] A. Ghosh, S. Pal, *Soft Computing Approach to Pattern Recognition and Image Processing*, World Scientific, pp.23-36, 2002.
[12] J. Lee, C. Lee, "The Neural-Fuzzy Control of a Transformer Cooling system", *International Journal of Advanced Culture Technology (IJACT)*, Vol.4, No.2, pp.47-56, 2016.
[13] S. Yoo, "Adaptive Thinning Algorithm for External Boundary Extraction", *International Journal of Advanced Culture Technology (IJACT)*, Vol.4, No.4, pp.75-80, 2016.