PROPOSED METHOD FOR IMAGE SEGMENTATION USING GRAPH THEORY

Balsam Hashim Kalleel and Prof. Dr. Kadhim M. Hashim
bлемhashim@gmail.com; Kadhimmehdi63@gmail.com
Computer science department, College of Education for Pure Sciences, University of Thi-Qar, Iraq

Abstract: This paper presents an image segmentation technique using graph tools for object detection. Graph theoretical systems have many good features among different segmentation schemes. It organizes the image elements into, mathematically and structural form, and makes the problem formulation more flexible, and the computation more efficient. In this paper, the work consists of two stages, in the first stage we apply the pixel-based labeling algorithm to the binary image, the algorithm works similarly to the eight connectivity labeling, but it is broader than it in terms of the search area, where two (horizontal and vertical) thresholds are first defined so that the search is in a block whose height is the vertical threshold and its width (2 * horizontal threshold), in the second stage the output of the first stage is mapping into an undirected weighted graph, in which each vertex represent region rather than pixel. We evaluate the results by comparing it with other method using (RI) parameter. We use 50 image from online source and image taken from Berkeley database.

Keywords: CCL, segmentation based-graph, graph.

1. INTRODUCTION

As a pre-processing step, image segmentation plays an important role in computer vision and image analysis [1], and since the image consists of pixels, it is difficult to extract high-level information for helping us to find and distinguish objects such as persons from these pixels, and here the need for segmentation is shown. A segmentation is an operation that divides the image into groups of pixels or areas that participate in certain properties such as density, texture and other features, and from these groups, important information can easily be extracted to use it in a wide range of applications such as medical application [2,3], object detection[4]. Various algorithms have been employed to extract this information from the images, and although there is a wide variety of image segmentation methods to choose from when performing segmentation, however, is Graph theoretical ideas which is one of the essential areas in mathematics used in structural models that lead to new inventions, and highly utilized by computer science applications [5] such data mining, image segmentation. Graph-based segmentation algorithms as mentioned in a survey presented by each of...
Vikramsingh R. Parihar, and Hamid Reza Boveiri in 2018[6], is one of the most popular approaches, the survey compared the graph-based segmentation approaches, thresholding-based segmentation approaches [7], segmentation approaches based on clustering [8], edge-based segmentation approaches [9], and segmentation approaches incorporating contours or Wavelets. Various papers related to graph theory have been studied.

In [10] defined a predicate for measuring the evidence for a boundary between two regions. Their algorithm produces segmentations that satisfy global properties. The specialty of this method is that it is able to preserve detail in low-variability image regions and ignore detail in high-variability regions. The running time of the algorithm, can be factored into two parts using the Berkeley Database. [11] [12] also based on Efficient Graph-Based Image Segmentation [10].

In 2018 [11], The author used the single-level discrete 2-D wavelet transform (DWT2) as preprocessing steps, then The graph-based segmentation is then performed on the filtered image based on the method stated in [10]. results are compared with the input image as well as ground truth data approaches is more preferable as the time differences are only of a few milliseconds. It is found that time taken for graph segmentation after DWT2 using (Haar) wavelets required less time for almost all the mentioned images. Used the performance evaluation parameters like Performance Ratio, execution time, PSNR. There is one runtime parameter for the approach. In [12]2019 Color and depth information used to segment scenes of RGB-D image. Multi-directional rotated bi-semi-Gaussian functions were used to calculate the dissimilarity (weight at each pixel) between two adjacent pixels. In the region merging stage, a novel method for computing, the merging threshold was presented by utilizing normal information in two adjacent regions. The experimental results of their method are compared with the state-of-the-art methods, using NYUv2dataset.

2. METHODOLOGY

In this thesis, we propose a method for image segmentation based graph theory that combines two techniques: connected component labeling (CCL) and region merging techniques. In our algorithm, the boundaries of objects preserved using canny edge detector; the work consists of three steps presented in (Fig.1), we first labeling the pixels of the image is attempting to cluster the points of binary image (initial segmentation) in order to reduce the number of vertexes which reduce the computation in the graph where each label represent vertex in a graph rather than pixels.
2.1 Image Pre-Processing
- Convert the image to binary.
- Canny Edge Detector.
- Apply the morphology dilation operation.

2.2 Connected Component Labeling (CCL)

For an $N \times N$ binary image, we use $I(x, y)$ to denote the value of a pixel at $(x, y)$ in the image, where $1 \leq x \leq N$, $1 \leq y \leq N$. We suppose that the value of foreground pixels is one and that of background pixels is zero. In our labeling algorithm we use two thresholds, vertical and horizontal thresholds, both define the area that represents the current pixel neighborhoods in which the search is modeled, similar to the eight-connectivity, but wider and more flexible, with better results.

2.2.1 Labeled Image

CCL algorithms have a clear and single result, in this algorithm after converts the image to binary, starts labeling its pixels. It is pixel-based labeling algorithm works similarly to the eight connectivity labeling, but it is broader than it in terms of the search area, where two (horizontal and vertical) thresholds are first defined so that the search is in a block whose height is the vertical threshold and its width $(2 \times \text{horizontal threshold})$ as the following steps:

Suppose $p$ is a pixel in an image, and begin scan from right to left and top to bottom.

$I(p)$=pixel value at position $p$.

$L(p)$=label assigned to pixel in location $p$.

1- Define specific two thresholds:
   - Vertical threshold, which refers to the vertical space between pixels to take the same label, i.e. the maximum vertical limitation.
   - Horizontal threshold, which refers to the horizontal space between pixels to take the same label, i.e. the maximum horizontal limitation.

2- If $I(p) = 0$, move to the next scanning position.
3- If $I(p) = 1$, scan all neighborhoods. (Neighborhoods that extend over the area of their height, the vertical
threshold, and its width to the right of the current pixel and its left is equal to the amount of the horizontal threshold for each side separately).

a- If \( I(p) = 1 \) and all neighborhoods = 0 then assign a new label to position \( p \), otherwise, \( l(p) = \) the label of the first pixel in scan that is not equal to zero.

b- If \( I(p) = 1 \) and there is more than one of the four neighbors (based on eight connectivity) have different labels then \( L(p) = \) the smallest label between those labels (representative label) and then propagate this label to eliminate the place of the largest label in all pixels that have a largest label.

The steps above can be illustrated with a simple example: We use the vertical threshold = 1, horizontal threshold = 2.

![Figure 2: Labeling using vertical threshold = 1, and horizontal threshold = 2](image)

In addition, the result on the image can be seen in figure (3), we use each image with different dimensions with a vertical threshold (vth) = 30, and horizontal threshold (hth) = 8.

*Note:* the dimensions of image affect the results.
We use image with (260,260), (126,126), (334,334) and (200,200) dimensions.

2.3. Undirected weighted graph construction

As we know, there is a wide variety of graphs; in this thesis, we use complete graph, in which, for each node, there are a links (edges) to all nodes in a graph. As each graph define as $G=\{V, E\}$ where $V$ is the set of vertices, and $E$ is the set of edges. Each vertex in the graph corresponds to a one connected component (represent one label) that results from the previous step (labeling algorithm) i.e. each node indicates to the group of pixels instead of one pixel, this reduces the computations. $V=\{v_1, \ldots, v_L\}$, where $L$ is the number of labels.

The weight of each edge represent the dissimilarity between two nodes, we first compute the average of intensity to image for each node.

$$\text{Av}(C_i) = \frac{1}{N} \sum_{k=1}^{N} I(x, y) \quad \text{................. (1)}$$

Where $\text{Av}(C_i)$ is the average of component $C_i$ that belongs to the set of vertices, which represent the object, $N$ number of pixels in each label (in each node), $I(x, y)$ represent the value of each pixel in gray image.

Then the weight of the edge between two vertices or components ($C_1$, $C_2$) can define as:

$$W(C_1, C_2) = |\text{Av}(C_1) - \text{Av}(C_2)| \quad \text{................. (2)}$$

2.4 Graph Cut Criteria

In order to combine two components, there are two conditions that must be fulfilled; figure (4) summarized this step.

The first condition:

$$W(C_1, C_2) = \begin{cases} 
\text{True} & \text{if } W(C_1, C_2) > \text{mint}(C_1, C_2) \\
\text{false} & \text{otherwise} 
\end{cases} \quad \text{........(3)}$$

This mean that the weight of edge between the component must be greater than mint ($C_1$, $C_2$) to satisfy the first merging condition, mint($C_1$, $C_2$) can be define as:

$$\text{mint}(C_1, C_2) = \min[\text{Av}(C_1) + S(C_1), \text{Av}(C_2) + S(C_2)] \quad \text{........(4)}$$
Where $z = 80$ is a constant, $|C_i|$ indicates to the number of pixels in component “i”.

The second condition: to check this condition, we must go through two steps:

First step: we need to calculate the center of area for each component (vertex)

Second step: calculate the Manhattan distance between the two components as:

\[ ED(C_1, C_2) = |x_1 - x_2| + |y_1 - y_2| \] \hspace{1cm} (6)

Where $(x_1, y_1), (x_2, y_2)$ the coordinates of the center of area for $C_1, C_2$.

When $ED(C_1, C_2) < 45$, then this condition is satisfy. If the two conditions satisfied then mean, the two components belong to the same object, and they are combined to form one vertex together in the updated graph, in other words, the two components will have the same label in labeling array after update it.

Algorithm 1: image segmentation using graph

| Input: labeled image, number of labels |
| Output: Final segmentation results |
| Step 1: building the undirected graph. |
| REPEAT |
| Step 2: compute the center of area for each vertex. |
| Step 3: compute the weight for each edge. |
| Step 4: Sort the weights in ascending order. |
| Step 5: check the two conditions for each edge. |
| If the conditions are met then the vertex that connected by the edge will merge and go back to the step 2. |
| UNTIL: scan all edges in a graph. |

Figure (4): region merge criteria
| Original image | 126*126 | 200*200 | 260*260 | 334*334 |
|---------------|---------|---------|---------|---------|
| ![CCL results](image1.png) | ![CCL results](image2.png) | ![CCL results](image3.png) | ![CCL results](image4.png) | ![CCL results](image5.png) |
| 1- 3 objects | 3 objects | 4 objects | 5 objects | 7 objects |
| ![Graph results](image6.png) | ![Graph results](image7.png) | ![Graph results](image8.png) | ![Graph results](image9.png) | ![Graph results](image10.png) |
| ![CCL results](image11.png) | ![CCL results](image12.png) | ![CCL results](image13.png) | ![CCL results](image14.png) | ![CCL results](image15.png) |
| 2- 5 objects | 5 objects | 5 objects | 5 objects | 5 objects |
3. EXPERIMENTAL RESULTS

Because, the labeling algorithm produced over-segmentation image, in a table (1), we will see the effect of the segmentation algorithm using the graph to reduce the number of objects(labels) compared with the labeling algorithm. The Graph algorithm works well, where, when the outputs of the CCL algorithm are good, the role of the Graph algorithm is limited to maintaining the good results without effect. Our method was compared with OD[4] and TBES[13] using Probability Rand-Index (RI) for evaluation.

Table (1): comparison the results of CCL algorithm and graph-based segmentation algorithm.

We find the best results obtained from image with dimensions (126*126), figure (5) view samples of the results in (126,126). Where the results in these dimensions are excellent and almost accurate with all the images used, in these samples where the original images contained two objects, our method also gave two objects, and the implementation time was parts of a second.
Figure(5): group(A) shows the results of CCL algorithm, and group(B) shows the results of graph-based segmentation algorithm.

Table(2): time of execution of the 5 image above

| Image num. | 126*126 | 200*200 | 260*260 | 334*334 |
|------------|---------|---------|---------|---------|
| 1          | 0.1877  | 0.1939  | 0.2108  | 0.2462  |
| 2          | 0.1884  | 0.1931  | 0.2086  | 0.2161  |
| 3          | 0.1880  | 0.2226  | 0.2250  | 0.2402  |
| 4          | 0.1799  | 0.1915  | 0.2331  | 0.2846  |
| 5          | 0.1741  | 0.2024  | 0.2112  | 0.2367  |

Table(3): cooperation our method, OD and TBES

| Image num. | TBES | OD |
|------------|------|----|
| 1          | 0.80 | 0.93 |
| 2          |      |    |
| 3          |      |    |
| 4          |      |    |
| 5          |      |    |

Table(4): Comparing the accuracy of segmentation in different dimensions using Dice similarity coefficient and Rand Index

| Image num. | 126*126 | 200*200 | 260*260 | 334*334 |
|------------|---------|---------|---------|---------|
| 1          | CCL     | 0.75    | 0.85    | 0.75    | 0.85    | 0.67    | 0.8    | 0.43    | 0.6    |
| graph      | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       |
| 2          | CCL     | 0.75    | 0.85    | 0.75    | 0.85    | 0.67    | 0.8    | 0.43    | 0.6    |
| graph      | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       | 1       |
| 3          | CCL     | 0.4     | 0.57    | 0.25    | 0.40    | 0.33    | 0.5    |        |        |
| graph      | 1       | 1       | 0.5     | 0.67    | 0.29    | 0.44    | 0.40    | 0.57    |        |
| 4          | CCL     | 0.67    | 0.8     | 0.5     | 0.67    | 0.18    | 0.30   |        |        |
| graph      | 1       | 1       | 0.67    | 0.8     | 0.67    | 0.8     | 0.33    | 0.5    |        |
| 5          | CCL     | 0.67    | 0.8     | 0.40    | 0.57    | 0.25    | 0.40   | 0.57    |        |
| graph      | 1       | 1       | 0.67    | 0.8     | 0.5     | 0.67    | 0.40    | 0.57    |        |

CONCLUSION

In this paper, we introduced a new method for image segmentation based on graph theory. Our segmentation algorithm consists of two stages, the first stage use CCL algorithm with two thresholds, which produce an over-segmentation image, and in second stage undirected weighted graph is constructed using the output of the first stage where each label(object) represent vertex in this graph and in this stage, two conditions must be satisfied to combine two components. The future work, use optimization algorithms to select the two thresholds, our method comparable with GB-RGB-D method.

5. REFERENCES

[1] (Digital Image Processing and Analysis, second edition, 2011 by Taylor & Francis Group, LLC)
[2] Wang, G., Li, W., Zuluaga, M. A., Pratt, R., Patel, P. A., Aerts, M., & Vercauteren, T. (2018). Interactive medical image segmentation using deep learning with image-specific fine tuning. IEEE transactions on medical imaging, 37(7), 1562-1573.
[3] Panchasara, C., & Joglekar, A. (2015). Application of image segmentation techniques on medical reports. International Journal of Computer Science and Information Technologies, 6(3), 2931-2933.
[4] Amrane, A., Meziane, A., & Boulkrinat, N. E. H. (2018, June). Object Detection in Images Based on Homogeneous Region Segmentation. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 327-333). Springer, Cham.
[5] DurgaPrasad, D., Muttipati, A. S., Snehadivy, M., & Kavitha, S. (2017). Applications of computer Science Based on Graph theory. International Journal of Engineering, Science and, 6, 1116-1122.

[6] Boveiri, V. R. (2018). A Survey and Comparative Analysis on Image Segmentation Techniques. In International Computer Science and Engineering Society (ICSES) — ISSN: 2467-292X. ITIPPR, Vol. 4, No. 4.

[7] Gurung, A., & Tamang, S. L. (2019). Image Segmentation using Multi-Threshold technique by Histogram Sampling. ArXiv preprint arXiv: 1909.05084.

[8] Li, Y., & Wu, H. (2012). A clustering method based on K-means algorithm. Physics Procedia, 25, 1104-1109.

[9] Reboüês Filho, P. P., da Silva Barros, A. C., Almeida, J. S., Rodrigues, J. P. C., & de Albuquerque, V. H. C. (2019). A new effective and powerful medical image segmentation algorithm based on optimum path snakes. Applied Soft Computing, 76, 649-670.

[10] P. F. Felzenszwalb and D. P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision, vol. 59, no. 2, pp. 167-181, 2004.

[11] V Threshold, an ICSES Book published by ITIPPR, vol. 4, no. 4, ISSN: 2467-293X, 2018. Ikram Singh R. Parihar and Ram Meghe, Image Segmentation Based on Graph Theory and Threshold.

[12] Pan, Xiong, et al (2019). Graph-based RGB-D image segmentation using color directional region. IEEE, pp. 2417-2421.

[13] Browet A., Absil P. A., Van Dooren P.: Community detection for hierarchical image segmentation. in Combinatorial Image Analysis. Springer, pp. 358-371 (2011).