International Journal of Electronics and Electrical Engineering

Volume 3 | Issue 4 Article 12

April 2015

ANALYSIS OF GRAPH CLUSTERING BASED NORMALIZED GRAPH CUT FOR IMAGE SEGMENTATION

K. RAJU
Dept of ECE ,KITS,Guntur-AP, raju.kolluri@gmail.com

DR.M.NARSING YADAV
Dept of ECE, MRIET, Maisammaguda, Hyderabad,, narsinghyadav@gmail.com

M. MARIYADAS
Electrical Sciences, GIET,Gunupur,Odisha., mariyadas@gmail.com

ANALYSIS OF GRAPH CLUSTERING BASED NORMALIZED GRAPH CUT FOR IMAGE SEGMENTATION

K. RAJU[1], DR.M.NARSING YADAV[2], M. MARIYADAS[3]

[1]Assoc.Professor, Dept of ECE ,KITS,Guntur-AP, [2].Professor, HOD (Dept of ECE), MRIET, Maisammaguda, Hyderabad, [3]Dean, Electrical Sciences, GIET,Gunupur,Odisha.
E-mail: raju.kolluri@gmail.com

Abstract: The humans have sense organs to sense the outside world. In these organs eyes are vital. The human eyes capture the light from the outside world and save the information as images in the brain. The human brain analyses the image data and gets the required information from the surroundings. Images are most prominent and easy way of representing a data. The art of representing information through the images is as old as the civilized man. Moreover the images can convey a clear data representation than the words or some other representation. Image segmentation is an old research topic, which has gained its importance in the past four decades. There are several previous methods for the segmentation. But there is no optimal solution for the judgment. This is because there is no specific benchmark for the judgment. In our project we propose a new method for the segmentation of an image called "The Normalized Graph Cut Segmentation". It is a global view concept which considers image as a graph model. The segmentation is done by using the similarity measurement technique. The problems of over segmentation and effect of noise can be overcome by this technique. The method is tested for various test cases like the landscape images, texture based images, high density feature based images and the performance of the algorithm has been tabulated.

Keywords: *Segmentation, Normalized cut, Texture, Landscape*

I. INTRODUCTION

Human beings are predominantly visual creatures: we rely heavily on our vision to make sense of the world around us. We not only look at things to identify and classify them, but we can scan for differences, and obtain an overall rough feeling for a scene with a quick glance. Humans have evolved very precise visual skills: we can identify a face in an instant; we can differentiate colors; we can process a large amount of visual information very quickly.

Although image processing can deal with changing scenes, we shall not discuss it in any detail in this text. For our purposes, an image is a single picture which represents something. It may be a picture of a person, of people or animals, or of an outdoor scene, or a microphotograph of an electronic component, or the result of medical imaging. Even if the picture is not immediately recognizable, it will not be just a random blur. An image may be defined as a two-dimensional function, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers the Image segmentation is a key step from image processing to image analysis, since a qualified segmentation is the prerequisite and base of the consequent processing such as object extraction, parameter measurement and object recognition. Image segmentation has always been an unsolved and challenging topic in image engineering for many years. Especially in recent years, with the appearance of various new theories and ideas, domestic and foreign researchers propose more and more segmentation methods or strategies, where a powerful tool, graph theory, has been introduced to image segmentation and brought lots of research fruits

Segmentation is an old research topic, which started around 1970, but there is still no robust solution toward it. There are two main reasons, the first is that the content variety of images is too large, and the second one is that there is no benchmark standard to judge the performance. Every technique has its own advantages also disadvantages, so it's hard to tell which one is better.

II. BACKGROUND STUDY

a) What is image processing?
Image processing involves changing the nature of an image in order to either:
1. Improve its pictorial information for human interpretation,
2. Render it more suitable for autonomous machine perception.

An image defined in the real world is considered to be a function of two real variables, for example, a(x, y) with a as the amplitude (e.g., brightness) of the image at the real coordinate position (x, y). An image may be considered to contain sub-images sometimes referred to as regions of interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition. The amplitudes of a given image will almost always be either real numbers or integer numbers. The latter is usually a result of a quantization process that converts a continuous range (say, between 0 and 100%) to a discrete number of levels. In certain image-forming processes, however, the signal may involve photon counting which implies that the amplitude would be inherently quantized. In other image forming procedures, such as magnetic resonance imaging, the direct physical measurement yields a complex number in the form of a real magnitude and a real phase. For the remainder of this book we will consider amplitudes as real or integers unless otherwise indicated.

b) Need for Segmentation

Image segmentation is used to separate an image into several “meaningful” parts. Image segmentation is an important image processing, and it seems everywhere if we want to analyze what inside the image. For example, if we seek to find if there is a chair or person inside an indoor image, we may need image segmentation to separate objects and analyze each object individually to check what it is. Image segmentation usually serves as the pre-processing before image pattern recognition, image feature extraction and image compression. Image segmentation is a technology which segments the image into the characteristics area and extracts the region of interesting in order to do the further analysis and processing. Image segmentation is a key step from image processing to image analysis, since a qualified segmentation is the prerequisite and base of the consequent processing such as object extraction, parameter measurement and object recognition. Image segmentation has always been an unsolved and challenging topic in image engineering for many years. Especially in recent years, with the appearance of various new theories and ideas, domestic and foreign researchers propose more and more segmentation methods or strategies, where a powerful tool, graph theory, has been introduced to image segmentation and brought lots of research fruits.

III. IMPLEMENTATION

The main objective of image segmentation is to extract various features of the image which can be merged or split in order to build objects of interest on which analysis and interpretation can be performed. In computer vision, Segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection).

a). Image Segmentation Based On Normalized Cut Frame Work

1) Introduction

In the normalized cut framework, we also model the image into a graph. We model each pixel of the image as a node in the graph, and set an edge between two nodes if there are similarities between them. The normalized cut framework is composed of two steps: similarity measurement and normalized cut process. The first step should be combined with feature extraction. The purpose of this step is to compute the similarity between pixels and this value is set as the weight on the edge.

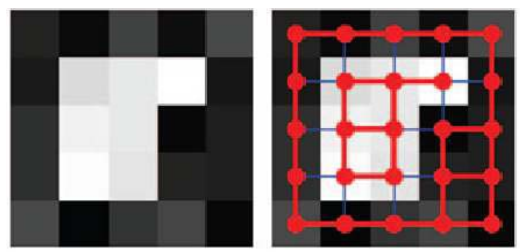


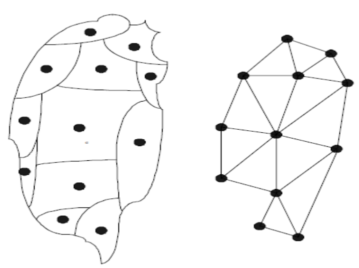
Figure 3: (a) is the original image, (b) this image has been modeled as a graph

This is a connected graph because each pixel could go through the edges to reach all other pixels else. The term “cut” means eliminating a set of edges to make the graph “unconnected”, and the cut value is the total weights on this set of edges. For example, if we eliminate all the blue edges in figure 3 then the nodes with white color will be “unconnected” to the nodes with dark color, and now we say the graph has been separate into two connected graph (the outside dark group and the inner white group). So, from the graph theory, the image segmentation problem is modeled as graph cut problem. But there are many kinds of cutting path we can adopt to separate the image into two part, we must have to follow some criterions. Remember that the weights on edges have the similarity meaning between pixels, so if we want to separate two pixels into two different groups, their similarity is expected to be small.

Three kinds of cutting criterions have been proposed in recent years: (1) minimum cut, (b) minimum ratio cut, and (c) minimum normalized cut, and the normalized cut has been proved to maintain both high difference between two segments and high similarities inside each segments. So in our project, we adopt the normalized cut framework.

2) Normalized Cut Framework:

The normalized cut framework is proposed by J. Malik and J. Shi. In their opinion, the image segmentation problem can be seen as a graph theory problem. Graph theory is an interesting math topic which models math problems into arcs (edges) and nodes. Although it's hard to explain graph theory in this project report, we give two practical examples to give readers more ideas about what it can do. In figure 4 a graph model of Taiwan map is presented, where we models each county as a node, and the edge between two nodes means these two counties are connected in the original map. This model could be used for coloring problems (give each county a color, while connected county should have different colors), or transportation flow problems in advanced. Each edge in the model could contain a value (weight), which could be used as flow or importance of it. This kind of graph is called "weighted graph", and is frequently adopted by internet researchers.



(a) (b)

Figure 4: (a) is a simplified Taiwan map, and (b) is the graph model of (a)

c) The formula for finding normalized cut:

In these two subsections, we'll present the mathematical derivation and algorithm implementation about how to find the normalized cut in a given image.

A graph G= (V, E) can be partitioned into two disjoint sets, A, B, A Ụ B = V, A ∩B = Ø by simply removing edges connecting the two parts. The degree of dissimilarity between these two pieces can be computed as total weight of the edges that have been removed. In graph theoretic language, it is called the cut.

$$\text{Cut}(A, B) = \sum u \in A, V \in BW(u,v) \quad (5)$$

The normalized cut then could be defined as

$$Ncut = \frac{CUT(A,B)}{ASSO(A,v)} + \frac{CUT(A,B)}{ASSO(B,V)} \quad (6)$$

Where asso(A, V) = Σu A,t vw(u, t), is the total connection from nodes in A to all nodes in the graph, and asso(B, V) is similarly defined. In the same spirit, we can define a measure for total normalized association within groups (a measure for similarities inside each group) for a given partition

$$\text{Nasso} = \frac{\text{asso}(A,A)}{\text{ASSO}(A,V)} + \frac{\text{asso}(A,B)}{\text{ASSO}(B,V)} \quad (7)$$

Here an important relation between Ncut(A, B) and Nasso(A, B) could be derived as followed:

$$\text{Ncut} = \frac{\text{CUT}(A,B)}{\text{ASSO}(A,v)} + \frac{\text{CUT}(A,B)}{\text{ASSO}(B,V)} \quad (8)$$

$$= \frac{\text{ASSO}(A,v) - \text{ASSO}(A,A)}{\text{ASSO}(A,V)} + \frac{\text{ASSO}(B,v) - \text{ASSO}(B,B)}{\text{ASSO}(B,V)} \quad (9)$$

$$= 2 - \text{Nasso}(A, B) \quad (10)$$

From this equation, we see that minimizing the disassociation between groups is identical to maximizing the association within each group.

IV. FEATURE EXTRACTION AND SIMILARITY MEASUREMENT

A similarity measure is a measure of correspondence between two images. If the similarity measure is maximal, the images are considered to be correctly aligned. Various image registration techniques utilize an image similarity measure to find the correct alignment of two images, for example: multi view, multi temporal or multimodal image registration.

The similarity measure may be used for image stitching or misaiming (multi view), motion detection or object tracking (multi temporal), or for information fusion (multimodal). Considering the problem of aligning two images, the simplest idea is to use the Euclidian distance between images. This idea can be easily extended to different correlation measures which are able to determine the similarity between images with an affine relationship between pixel intensities. However, the assumption of affine relationship between image intensity values does not always hold, especially in the case of multimodal image registration where different sensors are used for image acquisition of the same object, resulting in a complex relationship between pixel intensities.

Feature selection techniques provide three main benefits when constructing predictive models:

- Improved model interpretability,
- shorter training times, Enhanced generalization by reducing over fitting.

Feature selection is also useful as part of the data analysis process, as shows which features are important for prediction, and how these features are related. The simplest algorithm is to test each possible subset of features finding the one which minimizes the error rate.

a) Similarity Measurement:

To calculate the similarity between each pixel, we combine two measurements

- Feature similarity
- spatial similarity

1) Feature Similarity:
At first we use X2 distance to compute the distance between two histograms:

$$x^2(h_i,h_j)=\frac{1}{2}\sum_{l=1}^{L}\frac{[h_i(l)-h_j(l)]^2}{h_i(l)+h_j(l)} \quad (11)$$

Then we use exponential function as the distance measurement. As the distance getting far away, w_f, which means the similarity, will be smaller, where the of is a constant.

$$W_f = \exp\left(-\frac{x^2(h_i,h_j)}{\delta_f}\right) \quad (12)$$

2) Spatial Similarity:
But it is not sufficient if we care only the feature similarity, we also need to consider spatial similarity because general speaking if the pixels are far away, the similarity will be smaller:

$$W_p = \exp\left(-\frac{(x_i-x_j)^2}{\delta_p}\right), if \|x_i - x_j\| < r \quad (13)$$

Where of is a constant and r is a threshold if the distance too far away, we don't have to consider the similarity. Finally we have a similarity matrix. If the size of image is M-by-N, the size of similarity matrix will be (MN)-by-(MN)

V. IMPLEMENTATION OF ALGORITHM

Assume now we want to separate an image V with size M-by-N into two parts, we need to define two matrices: W and D, both of size (MN)-by-(MN). The matrix W is the similarity matrix with element w*, + as the similarity between the i - pixel and the j - pixel. The matrix D is a diagonal matrix and each diagonal element d* contains the sum of all the elements in the i - row in W. With these two matrices, finding the minimum normalized cut of image V into two parts A and B is equal to solve the equation as followed:

$$minNcut = m \quad (14)$$

where y is an (MN)-by-1 vector with each element indicating the attribute of each pixel into the two groups. Equation (4.4) could be further simplified into a general eigenvector problem as followed:

$$(D-W)Y=\lambda DY \quad (15)$$

a) Determine the vector Y:
There are several kinds of ways to define this threshold, for example, zero, mean value, and medium value among all the elements in y. In our Project, we use these three kinds of thresholds to get three different y. If an element in y is larger than the threshold, we set the element as 1, otherwise as –b. The value b is defined as:

$$b = \frac{\sum_{y>threshold} d_i}{\sum_{y<threshold} d_i} \quad (16)$$

We substitute the rebuilt y into equation (16) and choose the y with the minimum normalized cut value. Based on the two element values, we can separate pixels into two groups.

b) Procedure for Algorithm:
Figure 5 is the flowchart of the normalized cut framework. First we model the image into a graph and get the matrices W and D. Then we solve equation (15) and (16) to get the rebuilt y and separate the image into two segments. The normalized cut could only separate a segment into two parts in each iteration, so we need to recursively do the separation process to each segment. There is a diamond-shape block in the flowchart which serves the stopping mechanism of the recursive operation. For each segment, we check its area (number of pixels inside) and the further minimum normalized cut value, if the area is smaller than a defined threshold or the further minimum normalized cut value is larger than another defined threshold, the further separation process for this segment stops. The W and D for each segment could directly be extracted from the original W, so we don't have to rebuild it at each iteration. With this flowchart, we could solve the minimum normalized cut problem by Mat lab programs and implement the image segmentation operation.

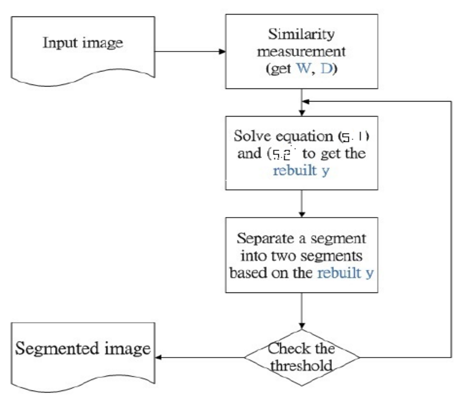


Figure 5: The flowchart of the normalized cut framework

VI. RESULTS

1. Test on Human face:

Input Image (a) (b) (c)

*Fig. Output images of 160*160 size*

(a) Edge computed, (b) Segmented image, (c)Pseudo color segmentation

Input Image **(a)** **(b)** **(c)**

Fig. Output images of 100*100 size
(a)Edge computed, (b) Segmented image, (c)Pseudo color segmentation

b) Test on building Image:

Input image **(a)** **(b)** **(c)**

Fig. Output images of 160*160 size
(a)Edge computed, (b) Segmented image, (c) Pseudo color segmentation

Input image ***(a)*** ***(b)*** ***(c)***

Fig. Output images of 100*100 size
(a)Edge computed, (b) Segmented image, (c) Pseudo color segmentation

Table 6.1 Computational time for 100*100 images,160*160 images

Image Size	Computation time	Image type
160*160	192.6263	Human Face
100*100	12.7234	Human Face
160*160	68.6892	Building
100*100	11.0361	Building

VII. CONCLUSION

From our experiments, we see that our methods could achieve the image segmentation purpose. For simple images with just a little texture inside, the result is quite good, and this method could also performs well on natural and landscape images. The above tests are based on the luminance or RGB based similarity measurement. We also find that the computation time of the eigenvectors is very slow. For a 100-by-100 image, it costs 20 seconds to get the final result, while it costs near 50 seconds for a 128-by-128 image. To solve this problem, we go through some papers but there is still no effective solution towards this problem.

REFERENCES

[1]. F. Sun, J. He, "A Normalized Cuts Based Image Segmentation Method," 2009 Second International Conference on Information and Computing Science

[2]. Shi. J, and Malik. J, "Normalized cuts and image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society, voI.22(8), 2000, pp.888-905.

[3]. Jitendra Malik, Serge Belongie, and Thomas Leung, "Contour and texture analysis for image segmentation", International Journal of Computer Vision, Kluwer Academic Publishers, vol.43(1), 2001, pp.7-27.

[4]. W. K. Pratt, "Image Segmentation," in Digital image processing, (4nd ed. Wiley, 2008, pp. 579-622.)

[5] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, "Hierarchy and adaptivity in segmenting visual scenes," Nature, vol. 442, no. 7104, pp. 810–813,2006.

[6] F. Sun, J. He, "A Normalized Cuts Based Image Segmentation Method," 2009 Second International Conference on Information and Computing Science.

[7] Shi. J, and Malik. J, "Normalized cuts and image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society, voI.22(8), 2000, pp.888-905.

[8] F. Sun, J. He, "The Remote-Sensing Image Segmentation Using Textons In The Normalized Cuts Framework," Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation, August 9 - 12, Changchun, China.

◈◈◈