Results of a Survey on Weighted Graph Cut Algorithms for Efficient Image Segmentation

Anmol Kapoor, Professor John R Gilbert

University of California, Santa Barbara

Table of Contents

Weighted Graph Cut Clustering for Efficient Image Segmentation

When it comes to distinguishing between the various parts of a scene, humans are adept at the task – determining where the road lanes end and the sidewalk begins is natural for the untrained eye to perceive and comprehend. If this were not the case, then the average human would not be a safe driver and we would see countless more car accidents and pedestrian fatalities. But when it comes to a computer processing the scene, the problem becomes more abstract. How exactly does the human brain process the different parts of a scene? How can we implement that logic such that a computer can understand the difference between road lanes and sidewalks? This problem is categorized as image segmentation. Image segmentation is a popular research topic in computer vision first developed in the early 1960s (IBM, 2002) and has various use cases in today's society. Efficient and accurate algorithms to differentiate between the parts of a scene are now depended on for human survival, especially with the advent of self-driving cars and computational image comprehension. Among the top performing image segmentation algorithms, graph cutting-based algorithms have been shown to meet these demands. This variety of groundbreaking segmentation algorithms were first theorized and developed in the late 1990s by Shi and Malik in their IEEE Normalized Cuts and Image Segmentation paper (Shi and Malik, 2000). Since then, numerous advances have been made in the field, and some algorithms are known to rival even human intelligence in terms of segmenting a scene (Martin and Fowlkes, 2001). To improve the models beyond just rivaling human intelligence, the algorithms efficiency must be improved. Two criterion for efficiency have been defined: first, the algorithm must perform all computations in running time nearly linear to the size of the images, and two, the algorithm must perform with valuable accuracy in various scenarios.

Naïve Clustering in Pattern Recognition

Of the various image segmentation algorithms, clustering on images is one of the traditional techniques, dating back to as far as 1993. One typical naïve algorithm that constitutes as image clustering is K-means clustering (Toshev and Daniilidis, 2004), which is a recursive algorithm that computes image clusters based on a certain point's distance to a refined average location. For the general and simple problems without refinement of the data, this use case works well and very fast, finishing in approximately O(n) running time. However, this naïve algorithm fails when two distinct segments of a graph are computed to have the same mean, but it is clear to the user that the groups are, in fact, very dissimilar. For this counterexample, consider a scatterplot on a 2-dimensional graph. Consider the approximation of two ring graphs as a dense circular scatter of points, where one smaller ring sits vertically and horizontally centered inside the twice as large ring. It can be observed that if we were to choose a K-value of 2 using the original data as described above, then the graphs would be partitioned across a straight line. This was not the intended clustering solution. Furthermore, the value of K is arbitrary and its value is not always as clear (Kumar et al, 2020). Another naïve algorithm for clustering is mean-shift clustering, which translates a predefined area around a graph until the maximal number of nodes can be represented within that graph. While this algorithm provides much better results than the K-means clustering, it runs much slower, in O(n^2) running time. The goal for using these graph-cutting based algorithms is in running time nearly linear to the number of pixels, so this O(n^2) running time is not satisfactory. Another algorithm, not as naïve, is the DBSCAN algorithm (Density-Based Spatial Clustering of Applications with Noise). This algorithm is based on mean shift clustering, except it has the advantage of not focusing on a specified area input, which allows the algorithm to perform well on distinct distributions of pixel data in an image. Furthermore, it is invariant to pixel outliers, a weakness of the K-means algorithm. The main drawback to this algorithm is that it is highly dependent on a continuous density distribution. As such, it does not perform well on good resolution images where the tail (or some other body part) of an animal is less dense graphically than the rest of the animal's body (Ester et al, 1996). One significant improvement to the model is that the model performs faster than the mean-shift algorithm, in time O(nlogn). One clustering algorithm that makes an improvement over the fast K-means algorithm is the Gaussian Mixture Model, which defines the clusters by a combination of mean and standard deviation. This offers better performance, and even solves the problem associated with a ring graph whose vertices are all within another ring graph. This algorithm performs in time nearly linear to its number of vertices in lower dimensions, but becomes exponentially slower as the dimensionality increases (Pinto and Engel, 2015). Another naïve clustering algorithm focuses on clustering in hierarchies and is implemented most commonly with the union find data structure on the condition of distance from cluster to cluster. This iterative algorithm combines points into small clusters and smaller clusters into larger clusters until the desired number of clusters has been achieved. This is the greatest advantage for this algorithm, but the main drawback is that the algorithm performs in a much worse running time: O(n^3) (Manning, et al., 2008). Other clustering algorithms focus on distinguishing regions based on their color, brightness, texture, and motion (Shi and Malik, 2000).

Early Clustering: Gestalt Clusters

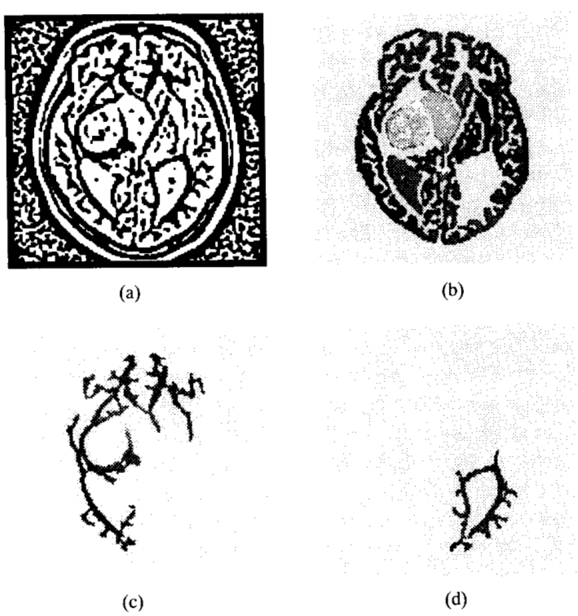
Gestalt clustering was first proposed by German psychologist Max Wertheimer and focused on the five key principles of how the brain's visual cortex operates: proximity, similarity, continuity, connectedness, and closure (The Albert Team, 2022). In 1971, Charles T. Zahn became among the first to implement these ideas into an algorithm into what he called detecting “Gestalt Clusters.” He focused on three of these principles in this algorithm: connectedness, proximity, and closure. Because these three principles depended on a nearest neighbor approach, Zahn chose to implement this algorithm using minimal spanning trees of edge weighted and linear graphs. He theorized that with the minimal spanning tree, it would be easier to describe algorithms that could show connectedness and closure. In constructing these minimal spanning trees, he theorized that the “hair” nodes of the graph had no purpose for the algorithm, and may even skew the results for segmentation. As such, his algorithm focused on removing these nodes and connecting those points that followed the closure ideal. Zahn defined these “hair” nodes as those having a degree of only one and connected to a node with degree at least 3 and determined that the closure ideal be fulfilled by connecting disconnected components that were close together but without edges in them. He would then use a minimal cut algorithm to distinguish between the connected components. Using these steps, his algorithm was able to describe the connectedness and closure qualities between graphs. Zahn reasoned that these edges he added for closure would not affect the output of the algorithm as most times the edges would anyways be deleted by the algorithm in segmentation. Zahn adds that proximity is easily implied by his usage of the minimax path, where if the MST T is an MST for graph G then for any two nodes x, y in G, the unique path in T is a minimax path. This minimax path focuses on minimizing the cost of the edge on the path with the largest weight. Intuitively, this makes sense. This theorem implies that the nodes x and y are as proximal to each other as the cost of this minimax path. So, a larger path distance implies a less proximal set of clusters, and a smaller path distance implies the nodes are close to each other (Zahn, 1971).

Cut Criterions

It is clear from Zahn's work that the method of using minimal spanning trees would be very useful for computing clusters. The only downside to this algorithm is the initial complexity in constructing the minimal spanning tree, which has a runtime complexity of O(nlogn), where n is the number of edges. Apart from that, running the algorithms on that tree becomes very efficient, running in nearly linear running time. The only other step left for computing these cuts is the criterion for splitting. It was clear from the previous section that, at least in terms of proximity, there is a floating-point scale describing the minimax path distance between any two nodes in the minimal spanning tree. The question then becomes what criterion should be used cutting the graphs and what delimiters should be used.

Optimal Graph Criterions Based on Maximum Flow

In 1993, Zhenyu Wu and Richard Leahy proposed such a data clustering criterion. They suggested, on top of the reduced tree defined earlier by Zahn, a new clustering algorithm based primarily on minimum cuts over the strong (highly weighted) edges while disregarding the strong edges that are more isolated. The clustering technique that they used was originally based on network flow theory, which focuses on directed trees where the edge weights may be distinct based on the orientation of the flow. Wu and Leahy find multiple advantages to this technique, one of which being that the subgraph partition created by the cut is the globally optimal partition for the adjacency graph. Other solutions, they argue, focus on the more iterative methods mentioned earlier, which typically find the locally optimal solutions, and these locally optimal solutions do not generalize well. Along with this reasoning, the iterative solutions may take a much longer time to compute for larger graphs, but because this maximum flow model is not an iterative solution, it can be run efficiently for larger graphs. As part of the implementation, they had to focus on two issues: the efficiency of the implementation scheme and the construction of the adjacency graph. They found that the efficiency of the implementation scheme is partially satisfied with the work of Gomory and Hu, the devisors of a multi-terminal maximum flow algorithm for undirected graphs. Their algorithm focuses on computing the minimum cuts and works well for graphs of moderate size, but for larger graphs the minimization algorithm performs linearly slower, which is not acceptable. Wu and Leahy then decided to add a point of thresholding, which allows them to forego the computation of each edge as long as it is greater than the largest minimum cut appropriate for the graph. This allowed the team of researchers to create an algorithm that uses this hierarchical implementation and disregard any larger-cost, or heavily correlated edges. With their algorithm, they were able to segment the image of a heart into three large components, shown below (Wu and Leahy, 1993).



Normalized Cuts and Image Segmentation

Whereas other papers implementing image segmentation focused on region-based merging and splitting algorithms, this paper focused on utilizing a graphical approach of the image where the weight on each edge is determined by some function of similarity between neighboring pixels, which are represented as nodes in the graph. This paper was one of the first to utilize approximate spectral analysis techniques in computing the segments in the image scene. In this paper, Shi and Malik focus on a new development in this pixel clustering problem, focusing on the issue that was presented in Wu and Leahy's paper, which was that it was favors isolated clusters. To address this issue, Shi and Malik focus on computing the cut cost as a function of the total edges in the graph, whereas Wu and Leahy had compared the total of the edge weights that were cut. Shi and Malik define this formulas for partitions A and B as:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

where,

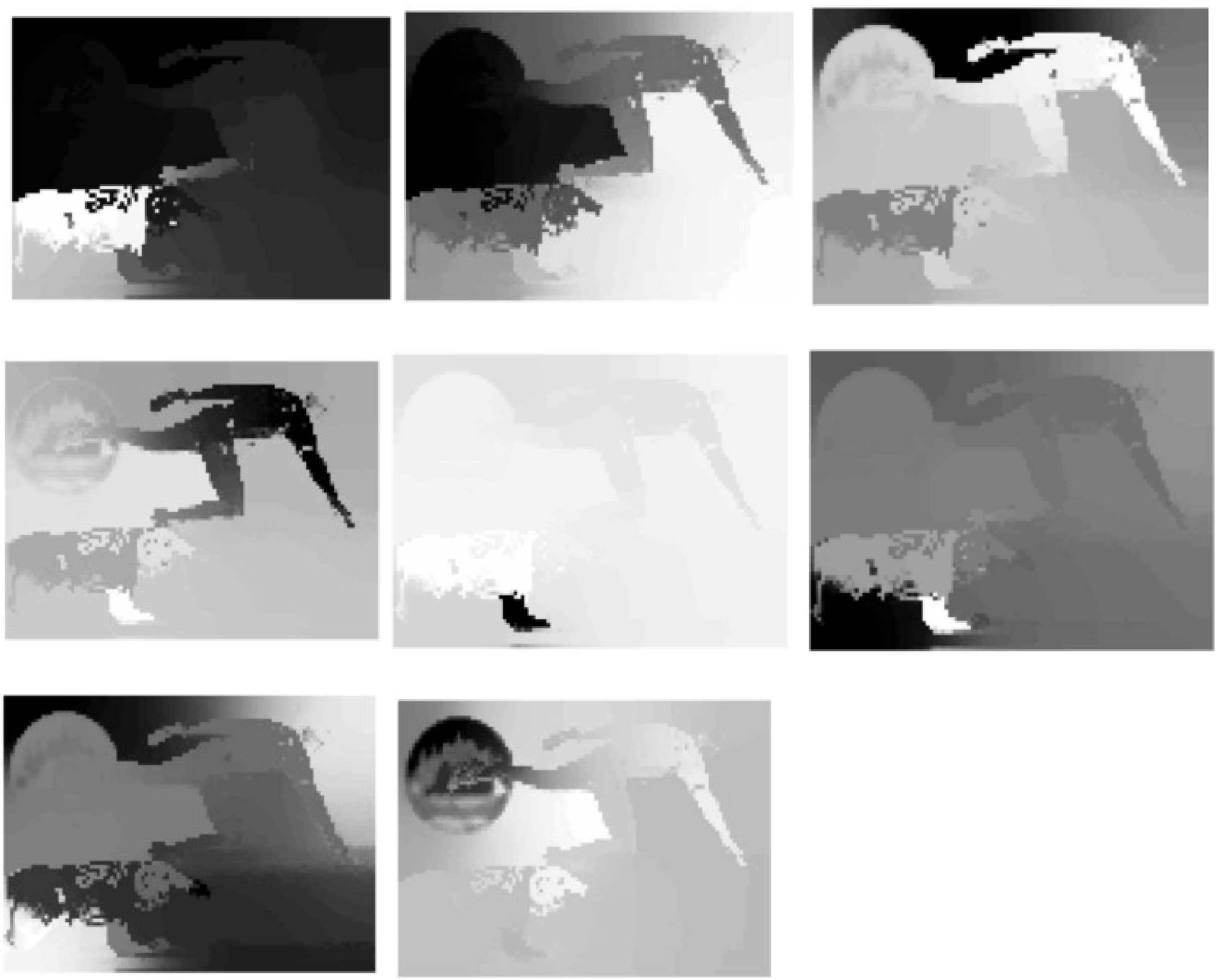
$$assoc(A,V) = \sum\nolimits_{u \in A, t \in V} \omega(u,t)$$

represents the total connection from nodes in A to all nodes in the graph. This normalized cut value, as a result, would no longer cut out the small and isolated points as in Wu and Leahy's algorithm. Through some optimizations and calculations, Shi and Malik find that their formula can be represented in terms of the Rayleigh Quotient:

$$\min_x Ncut(x) = \min_y \frac{y^T(D-W)y}{y^T Dy}$$

In this formula, they define D to be the diagonal matrix whose values correspond to the cost between 1 node and all other nodes in the graph, x to be a masking array (± 1), $y = (1+x) - \frac{k}{1-k}(1-x)$, k being the sum of all positively masked costs divided by the total cost, and W being Laplacian matrix of all the weights in the graph. As such, from the Rayleigh quotient, they define the eigenvalue system: $(D-W)y = \lambda Dy$. They test their theory on this grayscale image:

to produce the following results:



These eight resulting images are arrived at using different eigenvalues in the equation shown above. It is clear from these results that the Normalized cut is produce more generalized results than Wu and Leahy’s work. While this algorithm requires eigenvalue computation, it provides valuable results to the user (Shi and Malik, 2000).

References

Albert Team, T. (2019, December 4). Gestalt Principles: AP® Psychology Crash Course | Albert.io. Albert Resources; www.albert.io. https://www.albert.io/blog/gestalt-principles-ap-psychology-crash-course/

Arbelaez, P., Fowlkes, C., & Martin, D. (2007, 0 0). *Boundary Detection Benchmark: Algorithm Ranking*. Boundary Detection Benchmark: Algorithm Ranking; www2.eecs.berkeley.edu. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/algorithms.html

C. T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," in IEEE Transactions on Computers, vol. C-20, no. 1, pp. 68-86, Jan. 1971, doi: 10.1109/T-C.1971.223083.

International Business Machines (IBM). (2002, 0 0). *What is Computer Vision? | IBM*. What Is Computer Vision? | IBM; www.ibm.com. https://www.ibm.com/topics/computer-vision

Jalagam, Mahesh Kumar & Nanda, Rupalin & Rath, Rama & Rao, Gade. (2020). Image Segmentation using K-means Clustering. Journal of Advanced Science. 3700-3704.

Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000, doi: 10.1109/34.868688.

Manning, C. D., Raghavan, P., & Schütze, H. (2009, 0 0). *Time Complexity of HAC*. Introduction to Information Retrieval; nlp.stanford.edu. https://nlp.stanford.edu/IR-book/html/htmledition/time-complexity-of-hac-1.html

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (*KDD'96*). AAAI Press, 226–231.

Pinto RC, Engel PM (2015) A Fast Incremental Gaussian Mixture Model. PLoS ONE 10(10): e0139931. https://doi.org/10.1371/journal.pone.0139931

Toshev, A., Daniilidis, K. (2004). Graph Clustering and Image Segmentation: Based on Graph Based Image Segmentation Tutorial, She et.al, CVPR 2004 [PPT]. https://www.seas.upenn.edu/~cis580/lectures/580ncut.pdf.

Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pp. 1101-1113, Nov. 1993, doi: 10.1109/34.244673.