Analysis of Kruskal's Algorithm for Segmenting Image

Efron Manik

February 8, 2021

ISBN: 978-602-19937-2-9

Analysis of Kruskal's Algorithm for Segmenting Image

Efron Manik[1]

[1] Department of Mathematical Education FKIP UHN
efmanik@gmail.com

Abstract. Clustering / segmentation is widely used in the field of data mining. Even in image mining, clustering / segmentation image holds a very important role. Clustering / segmentation image is developed in a variety of approaches, among others: Region-growing approach, the approach boundary Region, Graph approach. One of the graph approach is the use of minimum spanning tree. Pixel of the image is seen as a point and the edge is seen as the difference in intensity for two adjacent points. Then we determine the minimum spanning tree using Kruskal's algorithm. Edge that weight is greater than a threshold discarded, so that will be formed several sub tree. Threshold used to cut the minimum spanning tree generated by Kruskal's algorithm, preferably in the form $\overline{c} + a\sigma$ with a value between 1 to 1.5, where $\overline{c}$ is the average weight and σ is standard deviation of minimum spanning tree. Suppose MST1, MST2 is different minimum spanning tree generated by Kruskal's Algorithm on graph G, c is weight on graph G, E(MST1) = $\{e_{11}, e_{12}, ..., e_{1p}\}$ is the set of edges on minimum spanning tree MST1, E(MST2) = $\{e_{21}, e_{22}, ..., e_{2q}\}$ is the set of edges on minimum spanning tree MST2, $c(e_{11}) \leq c(e_{12}) \leq ... \leq c(e_{1p})$, dan $c(e_{21}) \leq c(e_{22}) \leq ... \leq c(e_{1q})$. Then $p = q$, and $c(e_{1i}) = c(e_{2i})$ for each $i \in \{1, 2, ..., p\}$. Suppose $S_{T1} = \{T_{11}, T_{12}, ..., T_{1p}\}$, $S_{T2} = \{T_{21}, T_{22}, ..., T_{2q}\}$, respectively, is set of disjoint sub-tree of MST1, MST2 after the edge with the greater weight of the threshold is removed, and $P(T_{ij})$ is the set of points on sub tree T_{ij}, for i = 1, 2, dan j = 1, 2, ..., max$\{p,q\}$. Then $p = q$, and for each $s \in \{1, 2, ..., p\}$, there are $t \in \{1, 2, ..., p\}$ such that $P(T_{1s}) = P(T_{2t})$.

1 Introduction

Computer vision is a field of computer science that studies how computers can recognize objects in a digital image. Stereo vision is one of the areas of computer vision that discusses how to determine the three-dimensional coordinates of each pixel in the image. This field is very fast growing. Even Middlebury homepage (http: // paint. Middlebury.edu/stereo/) provides four pairs of stereo image with simple configuration to be contested, namely: Tsukuba, Venus, Teddy and Cones. One important step in any method used in stereo vision is clustering / image segmentation [1,2]

Clustering / segmentation is widely used in the field of data mining. Even in image mining, clustering / segmentation image has a central role. Clustering / segmentation image is developed in a variety of approaches, among others: Region-growing approach, the boundary approach, Graph approach. One approach is a Region-growing meanshift method. This method uses a series of vectors towards the converging point [3]. But the election of bandwidth greatly

 ISBN: 978-602-19937-2-9

affect the outcome of this segmentation. If the value of the greater bandwidth then the regions into smaller segments. Mean shift method still has weaknesses, namely: broader segment formed by adding bandwidth is not always a combination of segments for smaller bandwidth [4]. Use edge functions in Matlab is usually used to develop the border approach [5]. One of the graph approach is the use of minimum spanning tree. Pixels of the image is seen as a point while the edge is seen as the difference in intensity for two adjacent points. Subsequently, minimum spanning tree edge is determined and greater than threshold discarded [6].

This paper is a case study for image cones in Middlebury homepage. Issues to be studied are how the election of a minimum spanning tree for segmenting the image. How threshold election to get a good image segmentation.

We will discuss about the kinds of clustering and segmentation, and its use in Section 1. Section 2 discusses While spanning tree. Finally in Section 3 discusses the effect of using Kruskal's algorithm for image segmentation.

2 Reader Review

2.1 Graph

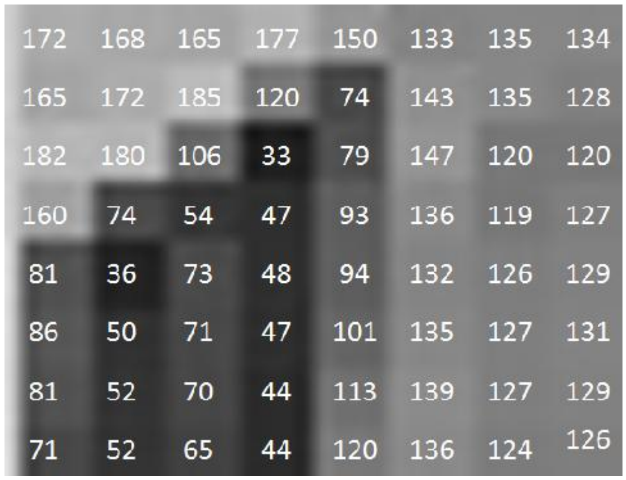
Definition 1. An **undirected graph** is a triple $(V, E, \vee)$, where V and E are finite sets and $\vee : E \to \{X \subseteq V : |X|=2\}$. A **directed graph** or **digraph** is a triple $(V, E, \vee)$, where V and E are finite sets and $\vee : E \to \{(v, w) \in V \times V : v \neq w\}$.By a **graph** we mean either an undirected graph or a digraph. The elements of V are called vertices, the elements of E are the edges.

Two edges e, e' with $\vee(e) = \vee(e')$ are called parallel. Graphs without parallel edges are called simple. For simple graphs we usually identify an edge e with its image $\vee(e)$ and write $G = (V(G), E(G))$, where $E(G) \subseteq \{X \subseteq V(G) : |X|=2\}$ or $E(G) \subseteq V(G) \times V(G)$. We often use this simpler notation even in the presence of parallel edges, then the "set" $E(G)$ may contain several "identical" elements. $|E(G)|$ denotes the number of edges, and for two edge sets E and F we always have $|E \cup F|=|E|+|F|$ even if parallel edges arise [7].For the purposes of the calculation we should not only rely on the presentation of the graph in the form of a set or graph. Presentation in the form of a matrix would be helpful, for example adjacency matrix, and edge matrix.

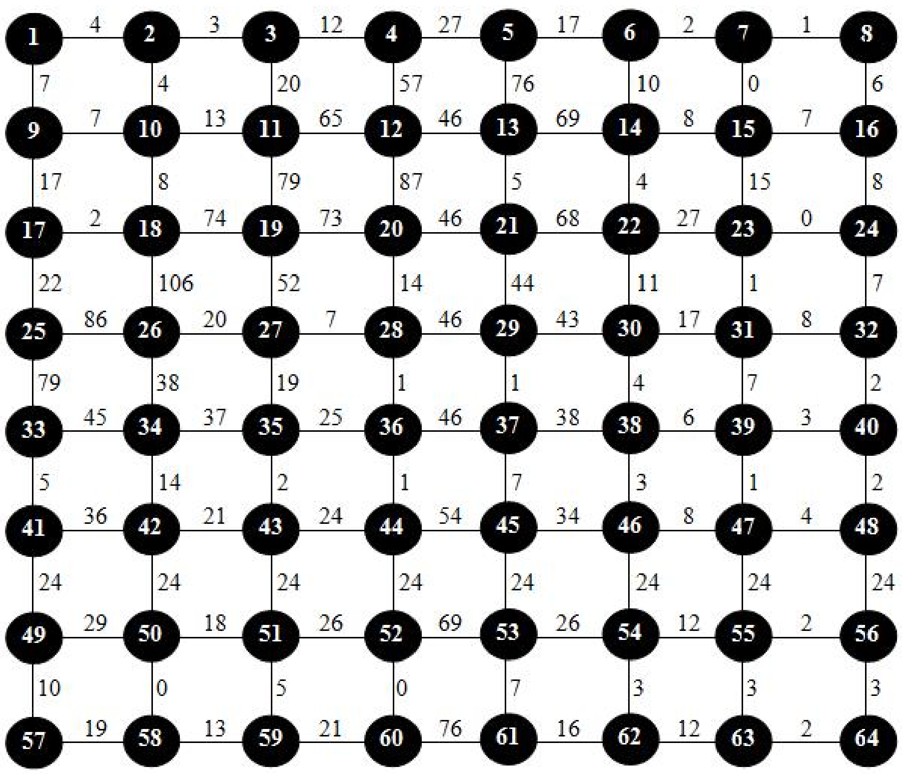
Graph used to present discrete objects and the relationships between these objects. Image is one example of the graph, where images can be viewed as a collection point / pixel, and the relationship of adjacent dots is seen as edge. The weights of these relationships can also be calculated by the difference in intensity of adjacent pixels. One way to make a graph of the image can be seen

 ISBN: 978-602-19937-2-9

in Fig.1. Pieces image and color intensity in Fig is taken from cones image in row 31 to 38 and column 436 to 443.



a. Pixel Intensity Values



b. Graph of Fig.1.a

Fig. 1. Formed Graph of Image

2.2 Minimum Spanning Tree (MST)

An undirected graph without a circuit (as a sub graph) is called a **forest**. A connected forest is a **tree**. A vertex of degree 1 in a tree is called a **leaf**. A **star** is a tree where at most one vertex is not a leaf. The following theorem is an existence and non-unique from the spanning tree.

Theorem 2. Every connected graph G has the spanning tree.

Theorem 3. Let G graph with n points, P incidence matrix, and J square matrix with all its elements 1. The amount of labeled spanning tree of a graph G is $\kappa(G) = \det(J + PP^t)/n^2$.

So we can make the spanning tree for each graph that is formed from the image. In this section, we consider the following two problems that are equivalent [7]:

Maximum Weight Forest Problem

Instance : An undirected graph G, weights $c : E(G) \rightarrow$ R.

Task : Find a forest in G of maximum weight.

Minimum Spanning Tree Problem

Instance : An undirected graph G, weights $c : E(G) \rightarrow$ R.

Task : Find a spanning tree in G of minimum weight or decide that G is not connected.

Proposition 4. The Maximum Weight Forest Problem and the Minimum Spanning Tree Problem are equivalent.

The following algorithm for the Minimum Spanning Tree Problem was proposed by Kruskal's Algorithm. It can be regarded as a special case of a quite general greedy algorithm. In the following let $n :=|V(G)|$ and $m :=|E(G)|$.

Kruskal's Algorithm

Input : A connected undirected graph G, weights $c : E(G) \rightarrow \mathbf{R}$.

Output : A spanning tree T of minimum weight.

1. Sort the edges such that $c(e_1) \le c(e_2) \le ... \le c(e_n)$.
2. Set $T := (V(G), \emptyset)$.
3. **For** $i := 1$ **to** m **do**:
   If $T + e_i$ contains no circuit **then** set $T := T + e_i$.

Theorem 5. Kruskal's algorithm produces a minimum spanning tree of a connected simple graph.

Proof. The proof consists of two parts [8, 9]. First, it is proved that the algorithm produces a spanning tree. Second, it is proved that the constructed spanning tree is of minimal weight.

(1) Spanning Tree: Let P be a connected, weighted graph and let Y be the sub graph of P produced by the algorithm. Obviously Y cannot form a circuit and Y cannot be disconnected, since the first encountered edge that joins two components of Y would have been added by the algorithm. Thus, Y is a spanning tree of P.

(2) Minimality: The proof is by contradiction. Assume that Y is not a minimal spanning tree and among all minimum weight spanning trees, we pick Y which has the smallest number of edges which are not in Y . Consider the edge e which was first to be added by the algorithm to Y of those which are not in Y_1. $Y_1 \cup e$ forms a circuit. Being a tree, Y cannot contain all edges of this circuit. Therefore this circuit contains an edge f which is not in Y . The graph $Y_2 = Y_1 \cup e \setminus f$ is also a spanning tree and therefore its weight cannot be less than the weight of Y_1 (since Y_1 is one of the minimal spanning tree with the smallest number of edges) and hence the weight of e cannot be less than the weight of f (i.e. $c(e) \geq c(f)$). However, the edge e is selected at the first step by Kruskal's algorithm which implies that it is of the minimal weight of all edges. (i.e. $c(e) \leq c(f)$). Thus the weight of e and f are equal ($c(e) = c(f)$), and hence Y_2 is also a minimal spanning tree. But Y_2 has one more edge in common with Y and Y_1, which contradicts to the choice of Y_1. □

2.3 Using Minimum Spanning Tree for Segmenting the Image

Peter [6] explains that the image of mining is more than just an extension of data mining for image domain. Image mining is a technique used to extract the direct knowledge of the image. Image segmentation is the first step in image mining. Minimum Spanning Tree-based Structural Similarity Clustering for Image Mining with Local Region Outliers (MSTSSCIMLRO) algorithm is used for segmenting the image. MSTSSCIMLRO given to detect anomalies (outlier). MSTSSCIMLRO algorithm uses Euclidean distance weighted to the side (edge), which is a key element in building a graph of the image. Image segmentation based MST is a fast and efficient method to produce a set of segments of an image. This algorithm uses a new cluster validation criteria based on geometric properties partitioning of the data set to find the right number of segments. The algorithm works in two stages. The first stage of the algorithm creates a number of clusters / segments optimal, where as the second phase of the algorithm further segment the optimal number of clusters / segments and detecting outliers local area.

Fig. 1 explains how to convert the image into a graph. Minimum Spanning Tree (MST), symbolized EMST1, constructed using Kurskal's algorithm. Furthermore, the calculated average of all edges contained in EMST1 ($\bar{c}$), and standard deviation (σ). Each side e_i with $c\ (e_i) > \bar{c} + \sigma$ removed from EMST1 and this has resulted in a set of disjoint sub-tree, for example: $S_T = \{T_1, T_2, ...\}$. Every T_i is seen as a cluster / segments

MSTSSCIMLRO algorithm does not require the users to select and try various parameters combinations in order to get the desired output. The benefit of the algorithm is to find similarity structures (segments) within clusters. Outliers have little or no influence, and may not be isolated as noise in the data. This algorithm does not assume fixed number of segments. According to how different pixels in the same cluster are allowed, the algorithm determines the numbers of segments through the processes. We do think that this is more natural way to segment image. All of these look nice from theoretical point of view, there is still some room for improvement for running time of the clustering algorithm [6].

3 Results and Discussion

Image used in this paper is the cones in Fig.2 taken from Middlebury homepage (http://cat.middlebury.edu/stereo/) with a gray color. Pieces image in Fig.1 taken from Fig.2 in row 31 to 38 and column 436 to 443. The square box on the top right area is a sample image that will be considered. This image is composed of 8 × 8 pixels. The intensity of each pixel can be seen in Fig.1.a and a graph that is formed can be seen in Fig.1.b.

Fig.2. Image of Cones

 ISBN: 978-602-19937-2-9

Graph in Fig.1.b written in the form of a matrix, where the figure in white in a circle stating point on the graph, and the number of black states the weight of the side that connects two points of them. Theorem 3 says that the spanning tree is not unique. Fig.3 is a minimum spanning tree that may be formed from the graph in Fig.1.b. There are 8 minimum spanning tree that may occur, namely: the product of the possibility of selecting two edge with weight 7 on the upper left area, 2 edge with weight 7 on the lower right area, and the two edge with weight 2 on the lower right area. With regard measures Kruskal's algorithm, minimum spanning tree results differ only by the difference in the selection of the same edge weight. So that the edge election does not reduce the amount of edge on the minimum spanning tree, and the sum of weights of all edges in the minimum spanning tree. These results will be written in Theorem 6.

Theorem 6. Suppose MST1, MST2 is different minimum spanning tree generated by Kruskal's Algorithm on graph G, c is weight on graph G, E(MST1) = $\{e_{11}, e_{12}, ..., e_{1p}\}$ is the set of edges on minimum spanning tree MST1, E(MST2) = $\{e_{21}, e_{22}, ..., e_{2q}\}$ is the set of edges on minimum spanning tree MST2, $c(e_{11}) \leq c(e_{12}) \leq ... \leq c(e_{1p})$, dan $c(e_{21}) \leq c(e_{22}) \leq ... \leq c(e_{1q})$. Then $p = q$, and $c(e_{1i}) = c(e_{2i})$ for each $i \in \{1, 2, ..., p\}$.

As a result of Theorem 6 is that the average weight of edges ($\bar{c}$) and standard deviation (σ), for all the possible minimum spanning tree, will not change. So for each minimum spanning tree elected in Fig .3, we obtain $\bar{c}$ = 8.5 and σ = 10.9.

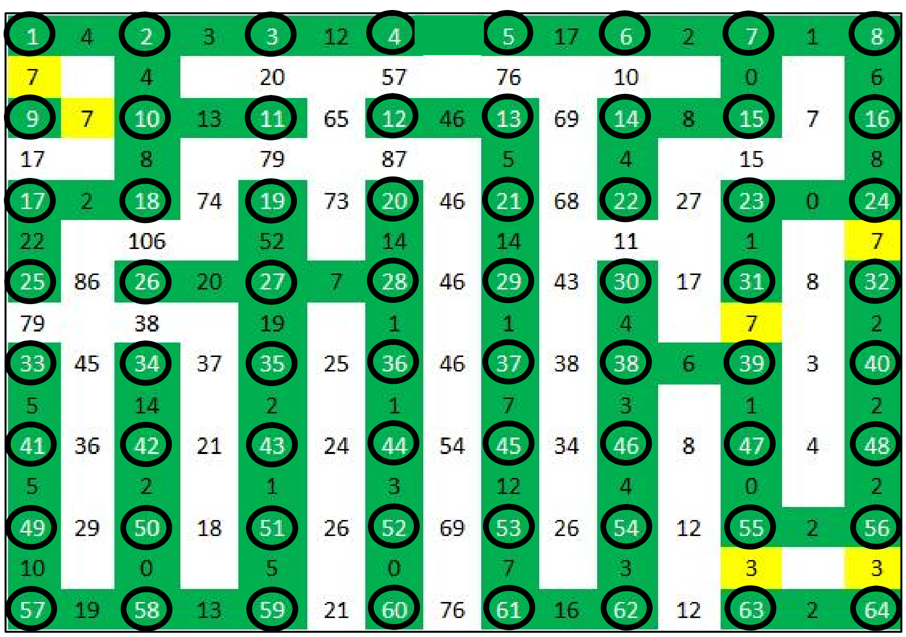


Fig.3. Minimum Spanning Tree of Fig.1.b Graph

 ISBN: 978-602-19937-2-9

Threshold is a linear combination of the average and standard deviation ($\bar{c} + a\sigma$), because the average and standard deviation does not change for each minimum spanning tree. Threshold used in this paper is the threshold that is not requested from the user. If we choose the threshold ($\bar{c} + \sigma$) equal to 19.4 then all edge that weighs more than 19.4 will be deleted. So edge (12, 13), (19, 27), (17, 25), and (26, 27) is removed from the minimum spanning tree. Fig (.4) is the result of the removal of all the edge and resulted in 7 sub tree (segment / cluster).

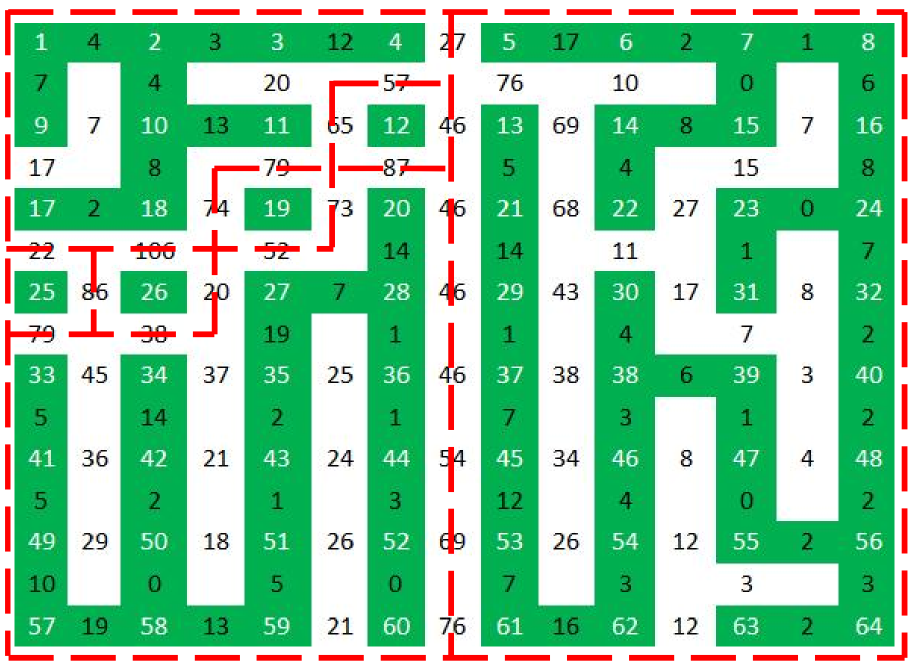
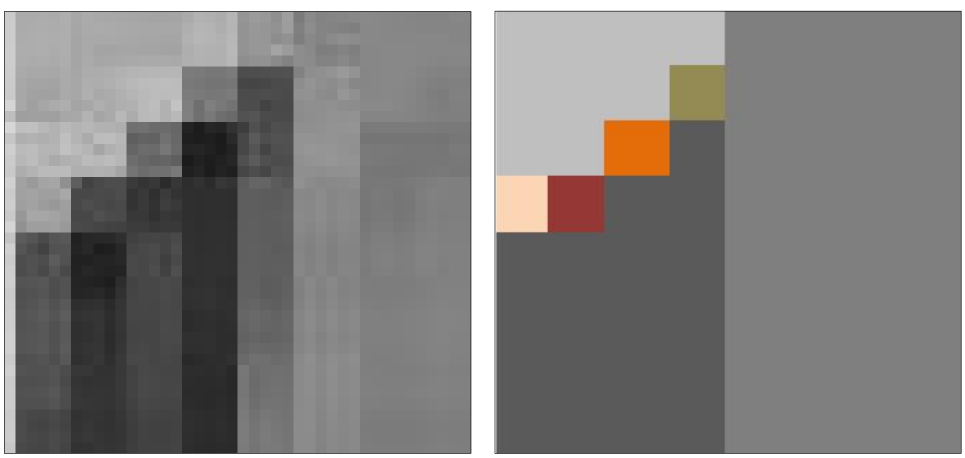


Fig.4. Seven Sub Tree or segment formed with Threshold 19.4

Sub tree (segments) which is formed in Fig.4 consists of seven sub-tree (segments) where 4 sub tree only has one point (pixel), one sub-tree (segments) by 9 points (pixels), 1 sub tree (segments) with 19 points (pixels), and a sub-tree (pixels) by 32 dots (pixels). Segments which have only one point is an outlier disturbing image segmentation. Collection points outlier is the boundary of a large segment. For each point / pixel at a sub tree is given the same color. The result can be seen in Fig.5.

We choose the threshold $\bar{c} + a\sigma$, where the value of $\bar{c} = 8.5$, $\sigma = 10.9$, a = 0, 0.5, 1, 1.5, 2, 2.5, 3 and 3.5. Selection of the threshold value greatly influences the amount of sub-tree or segments are formed. For a threshold equal to 8.5 (*a* = 0), the image is divided into 19 sub-tree, namely: 10 sub-tree with 1 pixel, 2 pixels sub tree with 2, 3 sub tree by 3 pixels, one sub-tree with four pixels, one sub tree with 6 pixels, 1 sub tree with 7 pixels, and 1 sub tree by 32 pixels. For a threshold equal to 13.9 (*a* = 0.5), the image is divided into 14 sub-tree, namely: 7 sub tree with 1 pixel, 1 sub tree with 2 pixel, 1 sub tree with four pixels, one sub-tree with a 5 pixel, 1 sub tree with 6 pixels, 1 pixel sub tree with 7, 1 sub tree by 9 pixels, and 1 sub tree by 32 pixels. For a threshold equal to 19.4 (*a* = 1), the image is divided into seven sub-tree, namely: 4 sub tree with 1 pixel, 1 sub tree by 9 pixels, 1 sub tree by 19 pixels, and 1 sub tree by 32 pixels. For a threshold equal to 24.8 (*a* = 1.5), the image is divided into 5 sub-tree, namely: 2 sub tree with 1 pixel, 1 sub tree with 10 pixel, 1 sub tree by 10 pixels, and 1 sub tree by 32 pixels. For a threshold equal to 30.3 (*a* = 2), the image is divided into 4 sub-tree, namely: 2 sub tree with 1 pixel, 1 sub tree by 10 pixels, and 1 sub tree by 42 pixels. For a = 2.5, and 3, is divided into 4 sub-tree the same as for a = 2. As for the threshold is equal to 46.6 (*a* = 3.5), the image is divided only into one sub-tree, namely: 1 sub tree by 64 pixels. With regard natural image, the selection of a value preferably between 1 and 1.5.



a. Natural Image b. Segment of Image

Fig.5. Image before and After Method

If we use the MSTSSCIMLRO algorithm to eliminate the edge that weight is greater than the threshold of the minimum spanning tree then we'll get some sub tree. With regard measures Kruskal's algorithm, the result of the minimum spanning tree will be distinguished by a selection edge, with the same weight, in each step. If all equally weighted edge is selected then the graph will have a circle that all points of the edge-edge lies in the circle, but if one edge is removed then for this step, the graph does not have a circle, but all of these points still lies in the same sub tree. If the threshold is greater than the weight of this edge, the sub-tree is not changed. If the threshold is smaller than the weight of the edge then all the edge with the same weight will be eliminated, and all this edge does not affect points on the sub tree. So the selection of the value threshold will not affect the location of the point on the sub tree that is formed. These results will be written in Theorem 7.

Theorem 7. Suppose MST1, MST2 is different minimum spanning tree generated by Kruskal's Algorithm on graph G, $S_{T1} = \{T_{11}, T_{12}, ..., T_{1p}\}$, $S_{T2} = \{T_{21}, T_{22}, ..., T_{2q}\}$, respectively, is set of disjoint sub-tree of MST1, MST2 after the edge with the greater weight of the threshold is removed, and $P(T_{ij})$ is the set of points on sub tree T_{ij}, for $i = 1, 2$, dan $j = 1, 2, ..., \max\{p,q\}$. Then $p = q$, and for each $s \in \{1, 2, ..., p\}$, there are $t \in \{1, 2, ..., p\}$ such that $P(T_{1s}) = P(T_{2t})$.

3 Conclusion

The conclusion of this paper is:

1. Threshold used to cut the minimum spanning tree generated by Kruskal's algorithm, preferably in the form $\bar{c} + a\sigma$ with a value between 1 to 2, where $\bar{c}$ is the average weight and σ is standard deviation of minimum spanning tree.
2. Suppose MST1, MST2 is different minimum spanning tree generated by Kruskal's Algorithm on graph G, c is weight on graph G, $E(MST1) = \{e_{11}, e_{12}, ..., e_{1p}\}$ is the set of edges on minimum spanning tree MST1, and $E(MST2) = \{e_{21}, e_{22}, ..., e_{2q}\}$ is the set of edges on minimum spanning tree MST2. Then $p = q$, and for each $i \in \{1, 2, ..., p\}$, there are $j \in \{1, 2, ..., p\}$ such that $c(e_{1i}) = c(e_{2j})$.
3. Suppose MST1, MST2 is different minimum spanning tree generated by Kruskal's Algorithm on graph G, $S_{T1} = \{T_{11}, T_{12}, ..., T_{1p}\}$, $S_{T2} = \{T_{21}, T_{22}, ..., T_{2q}\}$, respectively, is set of disjoint sub-tree of MST1, MST2 after the edge with the greater weight of the threshold is removed, and $P(T_{ij})$ is the set of points on sub tree T_{ij}, for $i = 1, 2$, dan $j = 1, 2, ..., \max\{p,q\}$. Then $p = q$, and for each $s \in \{1, 2, ..., p\}$, there are $t \in \{1, 2, ..., p\}$ such that $P(T_{1s}) = P(T_{2t})$.

References

1. Klaus, K., M. Sormann and K. Karner.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *Proceedings of the IEEE International Conference on Computer Vision*. (2006)
2. Yang,Q., L. Wang, R. Yang, H. Stewenius, and D. Nister.: Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern* Recognition. (2006)
3. Freedman, D. and P. Kisilev.: KDE Paring and a Faster Mean Shift Algorithm. *SIAM J. Imaging Sci.*, 3(4), 878–903. (2010).
4. Manik, E.: Pengaruh Bandwidth Terhadap Segmentasi Citra Digital Dengan Menggunakan Mean Shift. *VISI 18 (1) 43 -49*. (2010).
5. Uemura, T., G. Koutaki, and K. Uchimura.: Image Segmentation Based on Edge Detection Using Boundary Code. *ICIC International*. (2011)
6. Peter, S.J.: Minimum Spanning Tree-based Structural Similarity Clustering for Image Mining with Local Region Outliers. *IJCA Vol. 8(6)*. (2010)
7. Korte, B. and J. Vygen.: *Combinatorial Optimization: Theory and Algorithms, Third Edition*. Spinger. (2005)
8. Rosen, K. H.: *Discrete Mathematics and its applications, 6th ed*., McGraw-Hill, (2007)
9. Grimaldi R. P.: *Discrete and Combinatorial Mathematics, 5th ed*., Addison Wesley, (2004)