Minimum Spanning Tree-based Structural Similarity Clustering for Image Mining with Local Region Outliers

S. John Peter
Assistant Professor
Department of Computer Science and Research Center
St. Xavier's College, Palayamkottai
Tamil Nadu, India.

ABSTRACT

Image mining is more than just an extension of data mining to image domain. Image mining is a technique commonly used to extract knowledge directly from image. Image segmentation is the first step in image mining. We treat image segmentation as graph partitioning problem. In this paper we propose a novel algorithm, Minimum Spanning Tree based Structural Similarity Clustering for Image Mining with Local Region Outliers (**MSTSSCIMLRO)** to segment the given image and to detect anomalous pattern (outliers). In **MSTSSCIMLRO** algorithm we use weighted Euclidean distance for edges, which is key element in building the graph from image. **MST**-based image segmentation is fast and efficient method of generating a set of segments from an image. The algorithm uses a new cluster validation criterion based on the geometric property of data partition of the data set in order to find the proper number of segments. The algorithm works in two phases. The first phase of the algorithm creates optimal number of clusters/segments, where as the second phase of the algorithm further segments the optimal number of clusters/segments and detect local region outliers

General Terms: Graph Based Algorithm; Information retrieval ;

Keywords: Euclidean minimum spanning tree, Clustering, Cluster Separation; Segments, Eccentricity, Outliers;

1. Introduction

Image mining is a technique commonly used to extract knowledge directly from image. Image segmentation is the first step in image mining.

Image segmentation is closely related to the clustering problem. In Image analysis finding groups in data is very useful. We can find pixels with similar intensities ie., automatically finds regions in images. We can also find anomalous objects, which are present in the image. Segmentation can be viewed as partition a given image into regions or segments such that pixels belonging to a region are more similar to each other than pixel belonging to different regions. We also require that these regions be connected so regions consist of contiguous or neighboring pixels.
A large number of image segmentation techniques are available. These techniques are based on one of the following three approaches (i) clustering (ii) boundary deduction (iii) region growing. Image segmentation has the same relationship to image classification. In this paper we use Minimum Spanning Tree based clustering algorithm for segmenting images.

One of the best known problems in the field of data mining is clustering. The problem of clustering is to partition a data set into groups (clusters) in such a way that the data elements within a cluster are more similar to each other than data elements in different clusters [13].

We take a graph-based approach to segmentation. Let $G = (V, E)$ be an undirected graph with vertices $v_i \in V$, the set of elements to be segmented, and edges $(v_i, v_j) \in E$ corresponding to pairs of neighboring vertices/pixels. Each edge $(v_i, v_j) \in E$ has a corresponding weight $w(v_i; v_j)$, which is a non-negative measure of the dissimilarity between neighboring elements v_i and v_j. In the case of image segmentation, the elements in V are pixels and the weight of an edge is some measure of the dissimilarity between the two pixels connected by that edge (e.g., the difference in intensity, color, motion, location or some other local attribute). In the graph-based approach, a segmentation S is a partition of V into components such that each component (or region) $C \subseteq S$ corresponds to a connected component in a graph $G' = (V, E')$, where $E' \subseteq E$. In other words, any segmentation is induced by a subset of the edges in E. There are different ways to measure the quality of segmentation but in general we want the elements in a component to be similar, and elements in different components to be dissimilar. This means that edges between two vertices in the same component should have relatively low weights, and edges between vertices in different components should have higher weights.

Geometric notion of centrality are closely linked to facility location problem. The distance matrix D can computed rather efficiently using Dijkstra's algorithm with time complexity $O(|V|^2 \ln |V|)$ [30].

The *eccentricity* of a vertex x in G and radius ρ (G), respectively are defined as

$$e(x) = \max_{y \in V} d(x, y) \quad \text{and} \quad \rho(G) = \min_{x \in V} e(x)$$

The *center* of G is the set

$$C(G) = \{x \in V \mid e(x) = \rho(G)\}$$

C (G) is the center to the "*emergency facility location problem*" which is always contain single block of *G*. The

length of the longest path in the graph is called *diameter* of the graph *G*. we can define diameter D (G) as

$$D(G) = \max_{x \in V} e(x)$$

The *diameter* set of *G* is

$$Dia(G) = \{x \in V \mid e(x) = D(G)\}$$

In this paper, we consider outliers as points, (pixels) which are far from the most of other data (pixels). The proposed approach integrates Minimum Spanning Tree based clustering and structural similarity (density-based) approach for cluster pixels and detecting outliers. Here a vertex (pixel) is defined as an outlier if it participates in at most *TH* neighborhoods in **MST** graph, where threshold *TH is* control parameter. We classify a (pixel) vertex as outlier on basis of its *degree number* in the **MST** graph.

The paper is organized as follows. We review the related works on MST based clustering algorithms, image segmentation algorithms and different approaches of outlier detection in section 2. We formulize the notion of structure-connected clusters and describe the **MSTSSCIMLRO** algorithm in section 3. Finally in conclusion we summarize the strength of our method and possible improvements.

2. Related work

In image analysis, clustering can be used to find groups of pixels with similar gray levels, colors or local textures in order to discover the various regions in the image. A number of clustering techniques are available. In our approach we use Minimum Spanning Tree based clustering algorithm for clustering pixels.

Given a connected, undirected, weighted graph, $G = (V, E)$, where V is the set of nodes, E is the set of edges between pairs of nodes, and a weight $w(u, v)$ specifying weight of the edge (u, v) for each edge $(u, v) \in E$. A spanning tree is an acyclic sub graph of a graph *G*, which contains all vertices from *G*. The Minimum Spanning Tree (**MST**) of a weighted graph is minimum weight spanning tree of that graph. Several well established **MST** algorithms exist to solve minimum spanning tree problem [24, 20, 27]. The cost of constructing a minimum spanning tree is O $(m \log n)$, where m is the number of edges in the graph and n is the number of vertices. More efficient algorithm for constructing **MST**s has also been extensively researched [17, 13, 10]. These algorithms promise close to linear time complexity under different assumptions. A Euclidean minimum spanning tree (**EMST**) is a spanning tree of a set of n points in a metric space ($\mathbf{E^n}$), where the length of an edge is the Euclidean distance between a pair of points in the point set.

Zahn [34] proposes to construct MST of point set and delete inconsistent edges–the edges, whose weights are significantly larger than the average weight of the nearby edges in the tree. Zahn's inconsistent measure is defined as follows. Let *e* denote an edge in the **MST** of the point set, v_1 and v_2 be the end nodes of *e*, *w* be the weight of *e*. A *depth neighborhood* ***N*** of an end node *v* of an edge *e* defined as a set of all edges that belong to all the path of length *d* originating from *v*, excluding the path that include the edge *e*. Let N_1 and N_2 be the depth *d* neighborhood of the node v_1 and v_2. Let $\hat{W}_{N1}$ be the average weight of edges in N_1 and σN_1 be its standard deviation. Similarly, let $\hat{W}_{N2}$ be the average weight of edges in N_2 and σN_2 *be* its standard deviation. The inconsistency measure requires one of the three conditions hold:

1. $w > \hat{W}N_1 + c \times \sigma N_1$ or $w > \hat{W}N_2 + c \times \sigma N_2$

2. $w > \max(\hat{W}N_1 + c \times \sigma N_1, \hat{W}N_2 + c \times \sigma N_2)$

3. $$\frac{w}{\max(c \times \sigma N_1, c \times \sigma N_2)} > f$$

Where c and f are preset constant .All the edges of a tree that satisfy the inconsistency measure are considered inconsistent and are removed from the tree. This result in set of disjoint subtrees each represents a separate cluster.

Clustering by Minimal Spanning Tree can be viewed as a hierarchical clustering algorithm which follows the divisive approach. Clustering algorithm based on minimum and maximum spanning tree were extensively studied. Zahn [34] proposes to construct **MST** of point set and delete inconsistent edges – the edges, whose weights are significantly larger than the average weight of the nearby edges in the tree. Asano, Bhattacharya, Keil and Yao [1] gave optimal O $(n \log n)$ algorithm using maximum spanning trees for minimizing the maximum diameter of a bipartition. Asano, Bhattacharya, Keil and Yao also considered the clustering problem in which the goal to maximize the minimum inter-cluster distance. They gave a k-partition of point set removing the k-1 longest edges from the minimum spanning tree constructed from that point set [1]. The identification of inconsistent edges causes problem in the **MST** clustering algorithm. There exist numerous ways to divide clusters successively, but there is not suitable a suitable choice for all cases.

MST-based image segmentation is based on selecting edges from the graph, where each pixel corresponds to a node in the graph. Weights on each edge measure the dissimilarity between pixels. The segmentation algorithm defines the boundaries between regions by comparing two quantities- Intensity difference across the boundary and Intensity difference between neighboring pixels with each region. This is useful knowing that the intensity differences across the boundary are important if they are large relative to the intensity difference inside the at least on of the regions.

The min-max cut method [6] seeks to partition a graph $G = \{V, E\}$ into two clusters A and B. The principle of min-max clustering is minimizing the number of connections between A and B and maximizing the number of connections within each. A cut is defined the number of edges that would have to be removed to isolate the vertices in cluster A from those in cluster B. The min-max cut algorithm searches for the clustering that creates two clusters whose cut is minimized and while maximizing the number of remaining edges.
A normalized cut was proposed [29], which normalizes the cut by the total number connections between each cluster to the rest of the graph. Therefore, cutting out one vertex or some small part of the graph will no longer always yield an optimum.

Both min-max cut and normalized cut methods partition a graph into two clusters. To divide a graph into *k* clusters, one has to adopt a top-down approach, splitting the graph into two clusters, and further splitting these clusters and so on, until *k* clusters have been detected. There is no guarantee of the optimality of recursive clustering. There is no measure of the number of clusters that should be produced when *k* is unknown. There is no indicator to stop the bisection procedure.

For segmenting the image, the method proposed in [9] uses simple but effective modification of Kruskal's algorithm. This method addresses the problem of segmenting an image into regions by defining a predicate for measuring the evidence for boundary between two regions using a graph-based representation of the image. An important characteristic of the method is its ability to preserve detail in low-variability image region while ignoring detail in high-variability image region. The method [9] has several identified drawbacks. Firstly the internal difference is defined on the extreme values, which is not the accurate description of the components. Secondly the threshold function requires a user specified parameter *k* to control the size of the segmented regions. It is very difficult to choose an appropriate value for an expected segment size.

Ming Zhang et al. [26] proposed major improvement to [9]. They propose the method for sensor devices which are used for monitoring purpose. The method contributes to the method in [28] by re-defining the internal difference which is used to define the property of components. The internal difference is re-defined to give a more stable and accurate description of components. It also re-defines the threshold function which is the key element to determine the size of the components.

The article on color metric [32] suggests the use of weighted Euclidean distance in ***RGB*** color space. Explicitly, the methods in [9] and [28] use Euclidean method to calculate the edge weight in the graph. It is one of the key elements in the construction of the graph, which determine the segmentation result. However, if color image are considered, it is not just enough to consider the distance between two points. If we consider image containing red, green & blue components, there is need to associate some weight to each of these components in ***RGB*** color space. So weighted Euclidean function defined in [5] is used in the proposed algorithm. This is to ensure that a definite weight is associated with each of the red, green & blue components. We use the weighted Euclidean distance in order to compute the edge weight. The following equation is used for computing the edge weight.

$$|\Delta C| = \sqrt{(2 \times \Delta R^2 + 4 \times \Delta G^2 + 3 \times \Delta B^2)} \qquad (1)$$

where ΔR, ΔG *and* ΔB represent intensity the values of red, green & blue components in a two- dimensional space. 2, 4 & 3 represent the weight associated for red, green & blue components respectively [5].

Neighborhood-based Clustering (**NBC**) algorithm [36] proposed by Zhou S. G. et al is a good data clustering algorithm and can discover clusters of arbitrary shape and different densities using neighboring relationship among data points. To apply **NBC** to segment an image fast and efficiently Grayscale k-neighborhood based Density Factor (**GNDF**) [19] is introduced, which characterizes the local density of a gray's neighborhood in a relative sense.

There is no single universally applicable or generic outlier detection approach [23, 21]. Therefore there is many approaches have been proposed to deduct outliers. These approaches are classified into four major categories as *distribution-based, distance-based, density-based and clustering-based* [35].

Distribution-based approaches [11, 2] develop statistical models from the given data then apply a statistical test to determine if an object belongs to this model or not. Objects that have low probability to belong to the statistical model are declared as outliers. However, *distribution-based* approaches cannot be applied in multidimensional

dataset because of the univariate in nature. In addition, prior knowledge of the data distribution is required. These limitations have restricted the ability to apply these types of methods to large real-world databases which typically have many different fields and have no easy way of characterizing the multivariate distribution.

In the *distance-based* approach [21,22,23], outliers are detected using a given distance measure on feature space, A point *q* in a data set is an outlier with respect to the parameters *M* and *d*, if there are less than *M* points within the distance *d* from *q*, where the values of *M* and *d* are determined by the user. The problem in distance–based approach is that it is difficult to determine the *M* and *d* values.

In *Density-based* methods outlier is defined from local density of observation. These methods used different density estimation strategies. A low local density on the observation is an indication of a possible outlier. Brito et al [4] proposed a *Mutual k-Nearest-Neighbor* (**MkNN**) graph based approach. **MkNN** graph is a graph where an edge exits between vertices v_i and v_j if they both belong to each others k-neighborhood. **MkNN** graph is undirected and is special case of *k-Nearest-Neighbor* (**kNN**) graph, in which every node has pointers to its *k* nearest neighbors. Each connected component is considered as cluster, if it contains more than one vector and an outlier when connected component contains only one vector. Connected component with just one vertex is defined as an outlier.

Clustering-based approaches [25, 18, 16], consider clusters of small sizes as outliers. In these approaches, small clusters (clusters containing significantly less points than other clusters) are considered as outliers. The advantage of *clustering- based* approaches is that they do not have to be supervised.

The selection of the correct number of clusters is actually a kind of validation problem. A large number of clusters provides a more complex "model" where as a small number may approximate data too much. Hence, several methods and indices have been developed for the problem of cluster validation and selection of the number of clusters [31, 15, 14] based on the within and between-group distance.

3. MSTSSCIMLRO Algorithm

In this paper, we focus on simple, undirected and weighted graph. Let $G = \{V, E\}$ be a graph, where V is a set of

vertices/pixeles; and *E* is a set of pairs of distinct vertices/pixels, called edges. *W (u, v)* is the weight of the edge *(u, v).* The hierarchical method starts by constructing a Minimum Spanning Tree (**MST).** The weight of the edge in the tree is Euclidean distance between the two end points (vertices/pixels). Given an image the hierarchical method starts by constructing a Minimum Spanning Tree (**MST).** We named this **MST** as **EMST1.** Next the average weight $\hat{W}$ of the edges in the entire **EMST1** and its standard deviation σ are computed; any edge with $W > \hat{W} + \sigma$ or *current longest edge* is removed from the tree. This leads to a set of disjoint subtrees $S_T = \{T_1, T_2 \ldots\}$. Each of these subtrees T_i is treated as cluster/segment. The algorithm works in two phases. The first phase of the algorithm partitioned the **EMST1** into sub trees (clusters/regions/segments). The centers of clusters or regions or segments are identified using eccentricity of points. These points are a representative point for the each subtree S_T. A point c_i is assigned to a cluster/segment *i* if $c_i \in T_i$. The group of center points is represented as $C = \{c_1, c_2 \ldots \ldots c_k\}$. These center points $c_1, c_2 \ldots . c_k$ are connected and again minimum spanning tree **EMST2** is constructed is shown in the Fig 4. Xiaowei et al [33] proposed a method to find clusters, outliers and hubs based on undirected and un-weighted graph (Networks). The **MST** ignores many possible connections between the data patterns, so the cost of clustering/segmenting can be decreased. We modified the approach using Minimum Spanning Tree (**MST**). We propose a new algorithm; *Minimum Spanning Tree based Structural Similarity Clustering for Image Mining with Local Region Outliers (**MSTSSCIMLRO**)* for segmenting image. Here the number of edges between the vertices (pixels) is considerably reduced. So the performance of our approach has been improved.

Here, we use a cluster validation criterion based on the geometric characteristics of the clusters, in which only the inter-cluster metric is used. The **MSTSSCIMLRO** algorithm is a nearest centroid-based algorithm, which creates region or subtrees (clusters/regions/segments) of the data space. The algorithm partitions a set *S* of data of data *D* in data space in to *n* regions (clusters). Each region is represented by a centroid reference vector. If we let *p* be the centroid representing a region (cluster/segment), all data within the region (cluster) are closer to the centroid *p* of the region than to any other centroid *q*:

$$R(p) = \{x \in D \mid dist(x, p) \leq dist(x, q) \ \forall q\} \quad (2)$$

Thus, the problem of finding the proper number of clusters of a dataset can be transformed into problem of finding the proper region (clusters) of the dataset [8]. Here, we use the **MST** as a criterion to test the inter-cluster property. Based on this observation, we use a cluster validation criterion, called Cluster Separation (CS) in **MSTSSCIMLRO** algorithm.

Cluster separation (CS) is defined as the ratio between minimum and maximum edge of MST. ie.,

$$CS = E_{min} / E_{max} \quad (3)$$

where E_{max} is the maximum length edge of **MST**, which represents two centroids that are at maximum separation, and E_{min} is the minimum length edge in the **MST**, which represents two centroids that are nearest to each other. Then, the CS represents the relative separation of centroids. The value of CS ranges from 0 to 1. A low value of CS means that the two centroids are too close to each other and the corresponding partition is not valid. A high *CS* value means the partitions of the data is even and valid. In practice, we predefine a threshold to test the *CS.* If the *CS* is greater than the threshold, the partition of the dataset is valid. Then again partitions the data set by creating subtree (cluster/region). This process continues until the *CS* is smaller than the threshold. At that point, the proper number of clusters will be the number of cluster minus one. The *CS* criterion finds the proper binary relationship among clusters in the data space. The value setting of the threshold for the *CS* will be practical and is dependent on the dataset. The higher the value of the threshold the smaller the number of clusters would be. Generally, the value of the threshold will be > 0.8[8]. Fig 3 shows the *CS* value versus the number of clusters in hierarchical clustering. The CS value < 0.8 when the number of clusters is 5. Thus, the proper number of clusters for the data set is 4. Furthermore, the computational cost of *CS* is much lighter because the number of subclusters is small. This makes the *CS* criterion practical for the **MSTSSCIMLRO** algorithm when it is used for clustering/segmenting large dataset (image) and to detect outliers.

Our goal is both to cluster/segment graph optimally and to identify and isolate outliers. Therefore both connectivity and local structure is used in our definition of optimal clustering. Here we formulize the notion of structure-connected cluster/segments, which extends that of a density based cluster [7] and can distinguish good clusters/segments and outliers from image graph.

To detect the outliers from **EMST**, we use the *degree number* of points (objects/pixels) in the **EMST.** For any undirected graph *G* the *degree* of a vertex *v*, written as *deg (v),* is equal to the number of edges in G which contains *v*, that is, which are incident on *v* [12].

We propose the following definition for outliers based on **EMST,**

Definition 1: Given an **EMST** for a data set *S*, outlier is a vertex/pixel *v*, whose *degree* is equal to 1, with *dist (v, Nearest-Neighbor (v)) > TH.*

where *TH* is a threshold value used as control parameter. **EMST** is constructed from point set *S* (shown in Fig 1).Using graph partitioning method the subtrees (clusters) are created. For each of the subtrees vertices/pixels *v*, which have *degree* 1 are identified. Then we find *Nearest-Neighbor* for the above vertices *v*. The *distance* between the vertices *v* and its nearest neighbor vertex/pixel is computed. If the computed *distance* exceeds the threshold value *TH* then the corresponding vertices/pixels *are* identified as an outlier is shown in the Fig 2.

When scanning the **EMST,** the edges are ordered from smaller to larger lengths. Then we define the threshold as:

$$TH = max (L_i - L_{i-1}) * t \quad (4)$$

Where L_i is largest in the order and $t \in [0,1]$ is a user defined parameter.

The structure of a vertex/pixel in **EMST** can be described by its neighborhood. Let $v \in V$, the structure of v is defined by its neighborhood denoted by Γ(v)

$$\Gamma(v) = \{w \in V \mid (v, w) \in E\} \cup \{v\} \quad (5)$$

When a member of a cluster shares a similar structure with one of its neighbors, their computed structural similarity will be large. We apply a threshold ε to the computed structural similarity when assigning cluster membership, formulized in the following ε-neighborhood definition.

$$N_\varepsilon(v) = \{w \in \Gamma(v) \mid dist(v, w) \geq \varepsilon\} \quad (6)$$

When a vertex/pixel shares structural similarity with enough neighbors, it becomes a nucleus or seed for cluster. Such a vertex/pixel is called a core vertex/pixel. Core vertices are special classes of vertices/pixels that have a minimum of μ neighbors with a structural similarity that exceeds the thresholdε. From core vertices/pixels we grow the clusters/segments. In This way the parameters μ and ε determine the clustering/segmenting of graph. For a givenε. The minimal size of a cluster is determined by μ. Let $\varepsilon \in \Re$ and $\mu \in \aleph$, a vertex/pixel $v \in V$ is called a core w.r.t. ε and μ, if its ε-neighborhood contains at least μ vertices.

$$CORE_{\varepsilon,\mu}(v) \Leftrightarrow /N_\varepsilon(v)/ \geq \mu \quad (7)$$

We grow clusters from core vertices/pixels as follows. If a vertex/pixel is in ε-neighborhood of a core, it should be also in the same cluster/segment. They share a similar structure and are connected. This idea is formulized in the following definition of direct structure reachability.

$$DirREACH_{\varepsilon,\mu}(v,w) \Leftrightarrow CORE_{\varepsilon,\mu}(v) \wedge w \in N_\varepsilon(v) \quad (8)$$

Direct structure reachabiltiy is symmetric for any pair of cores. However, it is asymmetric if one of the vertices/pixels is not a core. The following definition is a canonical extension of direct structure reachability.

A vertex/pixel $w \in V$ is structure reachable from $v \in V$ w.r.t. ε and μ, if there is a chain of vertices/pixels $v_1, \ldots, v_n \in V, v_1 = v, v_n = w$ such that v_{i+1} is directly structure reachable from v_i, formally:

$$REACH_{\varepsilon,\mu}(v,w) \Leftrightarrow$$
$$\exists v_1, \ldots v_n \in V : v_1 = v \wedge v_n = w \wedge$$
$$\forall i \in \{1, \ldots n\text{-}1\} : DirREACH_{\varepsilon,\mu}(v_i, v_{i+1}) \quad (9)$$

The structure reachability is transitive, but it is asymmetric. It is only symmetric for a pair of cores. More specifically, the structure-reachability is a transitive closure of direct structure reachability. A non-empty subset $C \subseteq V$ is called a structure-connected cluster w.r.t ε and μ, if all vertices/pixels in C are structure-connected and C is maximal w.r.t structure reachability. A clustering P of graph $G = \{V, E\}$ w.r.t. ε and μ consists of all structure connected clusters/segments w.r.t. ε andμ in G.

Algorithm: MSTSSCIMLRO ( )

Input : *An Image* as point set *S*, ε, μ and *TH*
Output : optimal number of clusters/regions with *O* outliers

Let *e1* be an edge in the **EMST1** constructed from *S*
Let *e2* be an edge in the **EMST2** constructed from C
Let W_e be the weight of *e1*
Let σ be the standard deviation of the edge weights in **EMST1**
Let S_T be the set of disjoint subtrees of **EMST1**
Let *O* be set of outliers

1. Construct an **EMST1** from *S*
2. Compute the average weight of $\hat{W}$ of all the Edges from **EMST1**
3. Compute standard deviation σ of the edges from **EMST1**
4. $S_T = ø$; $C = ø$; $O = ø$;
5. **Repeat**
6. **For** each $e1 \in$ **EMST1**
7. **If** ($W_e > \hat{W} + \sigma$) or (current longest edge *e1)*
8. Remove *e1* from **EMST1**
9. $S_T = S_T \cup \{ T' \}$ // *T'* is new disjoint Subtree (regions)
10. Compute the center C_i of T_i using eccentricity of points
11. $C = U_{Ti} \in S_T \{C_i\}$
12. Construct an **EMST2** *T* from *C*
13. E_{min} = get-min-edge (*T*)
14. E_{max} = get-max-edge (*T*)
15. $CS = E_{min} / E_{max}$
16. **Until** $CS < 0.8$
17. **For** each T_i (cluster represented as **EMST2**) do
18. **For** $m = 1$ *to* $|T_i|$ *do*
19. **If** deg (v_m) == 1 **and** *dist* (v_m, *Nearest-Neighbor* (v_m)) > *TH* **then** $O = O \cup \{v_m\}$
20. **For each** unclassified vertex $v \in V$ in T_i **do**
21. **If** $CORE_{\varepsilon,\mu}(v)$ **then**
22. Generate new clusterID
23. **Insert** all $x \in N_\varepsilon(v)$ in to queue *Q*
24. **While** $Q \neq 0$ do
25. $y =$ **remove**(*Q*)
26. $R = \{x \in V \mid DirREACH_{\varepsilon,\mu}(y,x)\}$
27. **For** $x \in R$ **do**
28. **If** *x* is unclassified or non-member **then** assign current clusterID to *x*
29. **If** *x* is unclassified **then** **Insert** *x* into queue *Q*.
30. **Else**
31. label *v* as non-member
32. **Return** optimal regions/segments with *O*

Fig 1 shows a typical example of **EMST1** constructed from image point set *S*, in which inconsistent edges are removed to create subtree (clusters/regions/segments). Our algorithm finds the center of the each cluster/segment, which will be useful in many applications. Our algorithm will find optimal number of clusters/segments or cluster structures. Fig 2 shows the possible

distribution of the points in the three cluster structures with their center vertex/pixel as 5, 3 and 6.

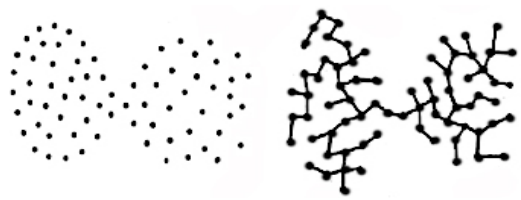
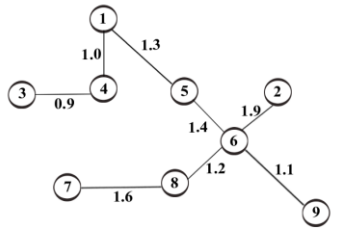


Fig 1**:** Pixels connected through points -EMST1

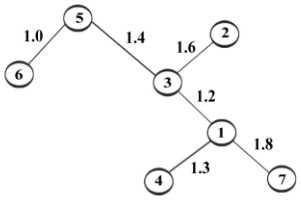


Vertex v	1	2	3	4	5	6	7	8	9
Eccentricity (v)	5.5	6.7	7.4	6.5	4.2	4.6	7.4	5.8	5.7
Degree (v)	2	1	1	2	2	4	1	2	1

Vertex v	2	3	7	9
Degree (v)	1	1	1	1
Nearest-Neighbour NN(v)	6	4	8	6
Dist(v,NN(v))	1.9	0.9	1.6	1.1

Center vertex = 5
Outlier vertex = 2

(a)

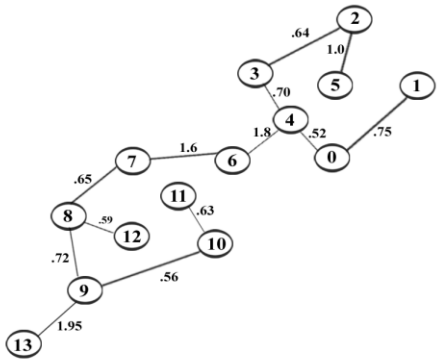


Vertex v	1	2	3	4	5	6	7
Eccentricity (v)	3.6	4.6	3.0	4.9	4.4	5.4	5.4
Degree(v)	3	1	3	1	2	1	1

Vertex v	2	4	6	7
Degree (v)	1	1	1	1
Nearest-Neighbour NN(v)	3	1	5	1
Dist(v,NN(v))	1.6	1.3	1.0	1.8

Center vertex = 3
Outlier vertex = 7

(b)



Vertex v	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Eccentricity (v)	7.24	7.99	8.06	7.42	6.72	9.06	4.92	5.74	6.39	7.11	7.67	8.3	6.98	9.06
Degree(v)	2	1	2	2	3	1	2	2	3	3	2	1	1	1

Vertex v	1	5	11	12	13
Degree (v)	1	1	1	1	1
Nearest-Neighbor NN(v)	0	2	10	8	9
Dist(v,NN(v))	.75	1.0	.63	.59	1.95

Outlier Vertex is 13

(c)

Fig 2: Three Clusters/regions (**EMST**) with center points 5 , 3 & 6 (outliers 2, 7 & 13)

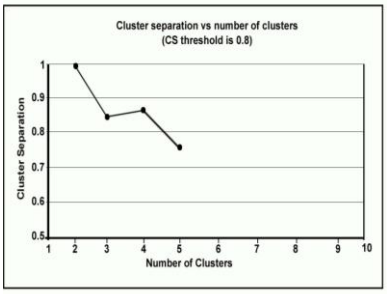


Fig 3: Number of Clusters vs. Cluster Separation

Our **MSTSSCIMLRO** algorithm works in two phases. The first phase (lines 1-16) of the algorithm finds optimal number clusters/segments with their center. It first constructs **EMST1** from set of point *S* (line 1). Average weight of edges and standard deviation are computed (lines 2-3). Inconsistent edges are identified and removed from **EMST1** to generate subtree *T'* (lines 7-9). The center for each subtree (cluster/region) is computed at line 10. Using the cluster/region /segment center point again another minimum spanning tree **EMST2** is constructed (line 12). Using the new evaluation criteria, optimal number of clusters/regions/segments is identified (lines 13-15). Lines 6-16 in the algorithm are repeated until optimal number of clusters/sements are obtained. The clusters/segments are well separated, shown in Fig 4. Here we describe the

MSTSSCIMLRO algorithm, which implements the search for clusters/segments and outliers. The **MSTSSCIMLRO** algorithm first generates optimal number of subtrees (clusters/segments). Then it visiting each vertex once to find structure-connected micro clusters/segments, and then it locate and identify outliers.

The Second Phase of the algorithm first identifies the outliers (lines 17-19). Then it performs one pass of an **EMST** and finds all structure-connected clusters/segments for a given parameters settings. At the beginning all the vertices/pixels are labeled as unclassified. The **MSTSSCIMLRO** algorithm classifies each vertex/pixel either a member of a cluster/segment or non-member. For each vertex/pixel that is not yet classified, **MSTSSCIMLRO** checks whether this vertex/pixel is a core (line 21). If the vertex/pixel is a core, a new cluster/segment is expanded from this vertex/pixel (lines 23-29). Otherwise the vertex/pixel is labeled as non-member (line 31). To find a new cluster **MSTSSCIMLRO** starts with an arbitrary core *v* and search for all vertices/pixels that are structure-reachable from *v* (line 25). New ClusterID is generated which will be assigned to all vertices found in (line 22). **MSTSSCIMLRO** begins by inserting all vertices/pixels in ε-neighborhood of vertex/pixel *v* in to a queue. For each vertex/pixel in a queue it computes all directly reachable vertices/pixels and inserts those vertices into queue which are still unclassified. This is repeated until the queue is empty.

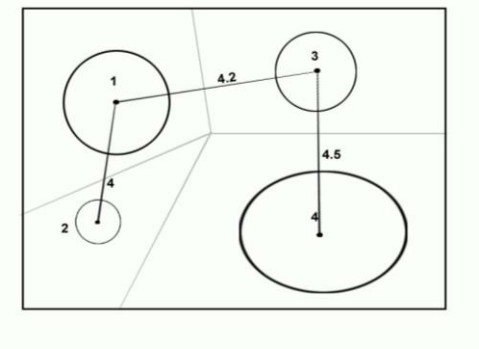


Fig 4: EMST2 From 4 region/cluster center points

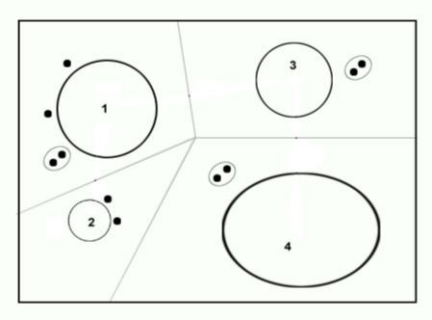


Fig 5. Four Clusters/regions with outliers as black spots

4. CONCLUSION

Our **MSTSSCIMLRO** algorithm gradually finds optimal number of clusters with segmentation for each cluster. Our 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. In this paper, we proposed **MSTSSCIMLRO** algorithm used to segments/ clusters image and detect outliers in graphs. The **MSTSSCIMLRO** clusters vertices /pixels based on their common neighbors. Two vertices/pixels are assigned to a cluster according to how they share neighbors. Our algorithm does not assume fixed number of segments. According to how different pixels in the same cluster are allowed, the algorithm determines the number 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. However from practical point of view, there is still some room for improvement for running time of the clustering algorithm. This could perhaps be accomplished by using some appropriate data structure. In this paper we consider only theoretical aspects for image segmentation. We hope the algorithm will produce better result for segmenting color images.

REFERENCES

[1] T. Asano, B. Bhattacharya, M.Keil and F.Yao. "Clustering Algorithms based on minimum and maximum spanning trees". *In Proceedings of the 4th Annual Symposium on ComputationalGeometry*,Pages252-257,1988.

[2] V.Barnett and T.Lewis, "Outliers in Statistical Data", *John Wiley*, 1994.

[3] M. Breunig, H.Kriegel, R.Ng and J.Sander, Lof: "Identifying density-based local outliers". *In Proceedings of 2000 ACM SIGMOD International Conference on Management of Data. ACM Press*, pp 93-104, 2000.

[4] M. R. Brito, E. L. Chavez, A. J. Quiroz, and J. E. Yukich. "Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection". *Statistics & Probability Letters*, 35(1):33-42, 1997.

[5] Deepthi Narayan, Srikanta Murthy K., and Hemantha Kumar G "Image Segmentation Based On Graph Theoretical Approach to Improve the Quality of Image Segmentation", *World Academy of Science, Engineering and Technology* 42, 2008.

[6] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A min-max cut algorithm for graph partitioning and data clustering", *Proc. of ICDM* 2001.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". *In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96), Portland, OR, pages* 291-316. *AAAI Press*, 1996.

[8] Feng Luo,Latifur Kahn, Farokh Bastani, I-Ling Yen, and Jizhong Zhou, "A dynamically growing self-organizing

tree(DGOST) for hierarchical gene expression profile" Bioinformatics,Vol 20,no 16, pp 2605-2617, 2004.

[9] Felzenszwalb P.F and Huttenlocher, 2004. "Efficient Graph-Based Image Segmentation", *International Journal of Computer Vision*, vol 59.

[10] M. Fredman and D. Willard. "Trans-dichotomous algorithms for minimum spanning trees and shortest paths". *In Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*,pages 719-725, 1990.

[11] Gath and A.Geva, "Fuzzy Clustering for the estimation of the Parameters of the components of Mixtures of Normal distribution", *Pattern Recognition letters*, 9, pp.77-86, 1989.

[12] Gary Chartrand and Ping Zhang "Introduction to Graph Theory", *Tata MgrawwHill, Paperback*-2008.

[13] S. Guha, R. Rastogi, and K. Shim. "CURE an efficient clustering algorithm for large databases". *In Proceeding of the 1998 ACM SIGMOD Int. Conf. on Management of Data , pp 73-84, Seattle, Washington*, 1998.

[14] A. Hardy, "On the number of clusters", *Computational Statistics and Data Analysis*, 23, pp. 83–96, 1996.

[15] D.Hawkins, "Identifications of Outliers", *Chapman and Hall*, London, ,1980.

[16] Z. He, X. Xu and S. Deng, "Discovering cluster-based Local Outliers", *Pattern Recognition Letters*, Volume 24, Issue 9-10, pp 1641 – 1650, June 2003.

[17] H.Gabow, T.Spencer and R.Rarjan. "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs", *Combinatorica*, 6(2):pp 109-122, 1986.

[18] M. Jaing, S. Tseng and C. Su, "Two-phase Clustering Process for Outlier Detection", *Pattern Recognition Letters*, Volume 22, Issue 6 – 7, pp 691 – 700, May 2001.

[19] Jundi Ding, SongCan Chen , RuNing Ma and Bo Wang, "A Fast Directed Tree Based Neighborhood Clustering Algorithm for Image Segmantation", Neural Information Processing , Lecture Notes in Computer Science, Vol 4233,pp 369-378, 2006.

[20] D. Karger, P. Klein and R. Tarjan, "A randomized linear-time algorithm to find minimum spanning trees", *Journal of the ACM*, 42(2):321-328, 1995.

[21] E. Knorr and R. Ng, "A Unified Notion of Outliers: Properties and Computation". *In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 219 – 222, August 1997.

[22] E..Knorr and R.Ng, "Algorithms for Mining Distance-based Outliers in Large Data sets", *Proc.the 24th International Conference on Very Large Databases(VLDB)*,pp.392-403, 1998.

[23] E.Knorr, R.Ng and V.Tucakov, "Distance- Based Outliers: Algorithms and Applications", *VLDB Journal*, 8(3-4):237-253, 2000.

[24] J. Kruskal, "On the shortest spanning subtree and the travelling salesman problem", *In Proceedings of the American Mathematical Society*, pp 48-50, 1956.

[25] A.Loureiro, L.Torgo and C.Soares, "Outlier detection using Clustering methods: A data cleaning Application", *in Proceedings of KDNet Symposium on Knowledge-based systems for the Public Sector*. Bonn, Germany, 2004.

[26] Ming Zhang , Reda Alhajj "Improving the Graph Based Image Segmentation Method", *Proceedings of the 18 th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, IEEE,2006.

[27] R. Prim. "Shortest connection networks and some generalization". Bell systems Technical Journal,36:1389-1401, 1957.

[28] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms", *in Proceedings Sixteenth IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004, Los Alamitos, CA, USA, IEEE Computer Society*, pp. 576–584 , 2004.

[29] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 22, No. 8, 2000.

[30] Stefan Wuchty and Peter F. Stadler. "Centers of Complex Networks". 2006

[31] S. Still and W. Bialek, "How many clusters? , An information-theoretic perspective", *Neural Computation*, 16, pp. 2483–2506, 2004.

[32] Thiadmer Riemersma, "Color Metric" Available at *http://www.compuphase.com/ cmetric.htm*

[33] Xiaowei Xu, Nurcan Yuruk Zhidan Feng and Thomas A.J. Schweiger, " SCAN: A Structural Clustering Algorithm for Networks", *SIGKDD*, San Jose, CA, US, 2007.

[34] C. Zahn. "Graph-theoretical methods for detecting and describing gestalt clusters", *IEEE Transactions on Computers*, C-20:68-86, 1971

[35] J. Zhang and N. Wang, "Detecting outlying subspaces for high-dimensional data: the new task, Algorithms and Performance", *Knowledge and Information Systems*, 10(3):333-555, 2006.

[36] Zhou, S., Zhao, J.: A Neighborhood-Based Clustering Algorithm. PAKD 2005, LNAI 3518 (1982) 361-371