Journal of Engineering and Applied Sciences 13 (Special Issue 1): 2274-2281, 2018
ISSN: 1816-949X

Graph Clustering for Images Based on Fractal Features

[1]Firas Miften and [2]Israa Hadi
[1]Faculty of Information Technology, University of Babylon, Hilah, Iraq
[2]Department of Computer Science, Faculty of Education for Pure Sciences,
University of Thi Qar, Nasiriyah, Iraq

Abstract: In recent years, there has been an increasing interest in emerging effective techniques for image clustering. Graph clustering algorithm partitions a set of vertices in graphs into smaller sets (clusters) such that vertices in the same set are related to each other than to those in other sets. This study focuses the problem whether or not the fractal is helpful for graph clustering images. The most of the study used fractal dimension to calculate the self-similarity. In this study, we present a new algorithm, based on matching rang and domain fractal to find self-similarity properties of the data sets which can be used for graph clustering. It suggests that using fractal for image graph clustering can get the effective results. And this study presents an algorithm that automatically finds the number of clusters based on shared neighbors among vertices. The proposed algorithm is able to efficiently find graph clustering partitions for whole graphs.

Key words: Fractal, self-similarity, fractal dimension, PIFS graph clustering, connectivity, density, automatic clustering, Jaccard similarity

INTRODUCTION

In image retrieval, similarity metrics are widely active. Several image processing approaches have been proposed lately to find solutions to the dilemma of content-based image retrieval. The fractal which stands for a modern theory suggested through last century, enriches a novel method to the problem clustering refers to dividing data into various sets of mini objects. In this regard, each set, known cluster, encompasses objects which appear similar in comparison with each other and different compared to entities of other sets (Maimon and Rokach, 2005; Kleinberg and Tardos, 2002).

Graph clustering in networks also called detecting communities, assumes a vital part in pattern recognition. Generally, it allows the distinguishing of groups of profoundly related vertices in a graph also, called clusters. The issue of graph clustering has gotten critical consideration amid the most recent years in view of its significance in different fields of science for example, the discovery of community in social networks, sensor networks, telecommunication and the web (Schaeffer, 2007).

Moreover, despite the multiplicity of clustering algorithms, a rarity can consequently find groups without the details of the sum of groups. Automatic graph clustering algorithms, ready to characterize independent alone the totality of groups, assume a vital part in the process of analyzing data since such groups permit a more productive use of clustering algorithms for the sake of applying to a dataset without earlier learning of the information adaptation. In this way, the examination of novel clustering algorithms ready to manage graph clustering issues and besides to identify automatically the collection of groups as a critical research matter.

In conventional clustering of sets of data, the way of distance measure can basically be based on the identification of attribute, e.g., Euclidian distance comparing the two attributes. As opposed to the current approach, graph clustering categorizes the vertex closeness depending upon connectivity, neighborhood similarity, attribute or contextual similarity. Many current algorithms of graph clustering regard the topological construction of a graph to fulfill the durable interior construction. This approach incorporates clustering based upon max flow min-cut problem (Andersen and Lang, 2006; Xu *et al*., 2007) normalized cut (Shi and Malik, 2000), structural density and modularity (Ino *et al*., 2005; Newman, 2004). Such methods divide the classes of nodes into variable groups as well as gauge the cost of edge cut, i.e., sum of edges relating vertices in various groups or edge cost relying on the connected weights. Such methodologies segment the order of vertices in different collections and gauge the cut cost edge, i.e., edges number interfacing vertices in various gatherings or edge cost in view of the related weights.

Corresponding Author: Firas Miften, Faculty of Information Technology, University of Babylon, Hilah, Iraq

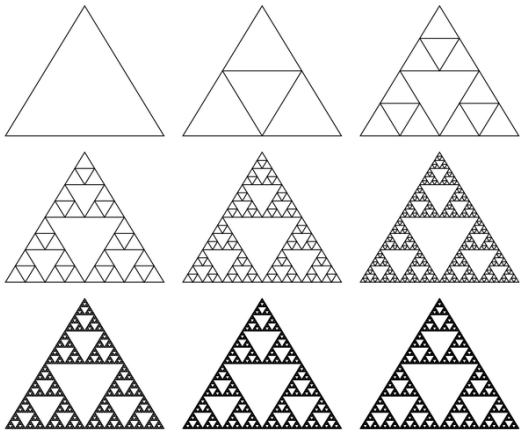

Fig. 1: Sierpinski triangle illustrates self-similarity

The mathematical description of fractal has been clarified by authority of Benoit Mandelbrot which realizes a fractal is a collection that the dimension of Hausdorff Besicovich severely outreaches this topological dimension. Though it represents a very abstract explanation. In general it is possible to identify the fractal as a fragmentized geometric form which may be distributed into segments where each part is (at the very least nearly) a reduced-size version of that whole. Fractals may be seen as in general self-identical as well as independent of scale (Leodkin *et al*., 2009). The conception of self-similarity in which an entity seems to be similar to various scales or objects whose small pieces resemble the whole as for example this is the Sierpinski Triangle in Fig. 1 illustrate self-similarity. In this mathematical object, each little piece is an exact smaller copy of the whole object (Mandelbrot, 1982; Kaye, 2008; Mon, 1986).

Image similarity measurement is the main process to distinguish the place through measuring the image similarity. The basic to measure the image similarity is to construct a vector or matrix which can label the individual characteristic of the image and recognize it from others. Most researchers developed techniques to mining images based on a fractal theory which is used the fractal dimension as the features. The extracted features are used as input to mining algorithms.

Shih (2014) extracted fractal dimension as features to cluster images. He suggested that using the dimension of fractal for image sequencing can develop the sufficiency in content-based image retrieval. Wenping and Yingde (Liu and He, 2010) sed the dimension of local fractal in the form of features to recognized the dissimilarity between on the one hand the tree leaf and on the other hand natural background. Liu *et al*. (2014) introduced fractal theory as image properties extraction. Fuzzy sequencing is used with neural network for processing characteristics. Tasoulis *et al*. (2014) present a classification tool for the image of computer-assisted based on fractal as well as fuzzy sequencing for the quantification of rate of the IPF (Idiopathic Pulmonary Fibrosis) in images. Wein and Blake (1996) used the technique of clustering operation on image domain mass with the clusters constituted by the use of k-d trees.

As regards the approach developed to treat clustering of graph node (Nawaz *et al*., 2015) it introduces the measure of Collaborative Similarity (CSM) aiming at clustering of intra-graph. Instead of the different paths, CSM depends on the strategy of shortest path to clarify the relevance of structure as well as semantics between vertices. Thus, the method surveyed in (Ren *et al*., 2015) suggests the name of congruent approximate graph Clustering (CAC) which may keep on the notion of non negativity severely and may arrive the orthogonality definitely through congruency approximation. On the other hand, the technique given by Van Dongen, concerns the arbitrary-pair attributes of vertices. Consequently, the values of the similar attribute are gathered under either specific partition or cluster. As such it stands as a sufficient way of graph summarization depending upon OLAP processes. As for the first process, known as SNAP it yields a summary graph via. collecting nodes by means of the node attributes and connections of the user-selected node. Concerning the second process, in k-SNAP one it further permits users to override summary resolution. In order to arrive better analogous of graph summarization to OLAP processes, vertices partition has taken place relying on their feature and then initiating summaries whereas ignoring the connectivity.

Partitioned Iterated Function Systems (PIFS): Fractal image compression used PIFS which is applied to a single image. Suppose we are dealing with the grayscale image I that has size W by W. The image is partitioned into n non-overlapping block size b by b is named as range blocks be represented by $R = \{R_1, R_2, ..., R_n\}$. Let D be the set of all possible blocks in the same image 1 which overlapping block of size 2b by 2b is named as domain blocks be represented by $D = \{D_1, D_2, ..., D_m\}$. Now for each range block R_i must find a suitably matched domain block D_j then save transformation parameters, thus, obtained is called the fractal codes of image 1.

Terminology and definitions: For the sake of discussion simplicity it is necessary to put forth this symbol: a weighted an undirected, a graph G consists of an ordered

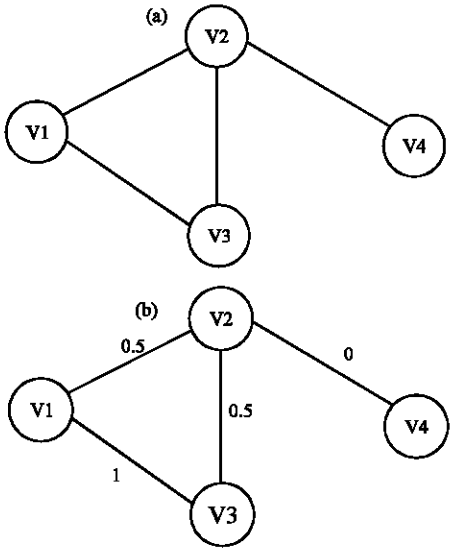

Fig. 2: Similarity among vertices: a) Unweighted graph and b) Construct aweighted graph by Jaccard similarity

pair G = (V, E) where V stands for a class of vertices and E represents a class of edges. In addition, the matrix of similarity (matrix of affinity) of G graph on n vertices can be expressed by $W = (w_{ij})$ i, j = 1, …, n $\in R_{n\times n}$. The positive entry w_{ij} in W refers to vertex i while vertex j seems related together a weighted edge. If $w_{ij} = 0$, it indicates the i as well as j vertices that cannot be related by the edge. Moreover, the matrix of similarity W stands for symmetric for undirected graphs:

$$\mathrm{Sim}(X,Y)=\frac{|X\cap Y|}{|X\cup Y|} \quad (1)$$

Jaccard (1901) similarity refers to coefficient measure presented in Eq. 1. It is, usually speaking, numerously used and acceptable in the area of data mining (Everitt *et al*., 2001). Because of its simplicity it is applied in several areas to detect the relevance between the objects. In this research, it will be used to redefine edge weights between vertices via. similarity of Jaccard (Fig. 2).

Similarity among v1 and v1 by utilizing the neighborhood of these two vertices. Worded differently, it stands for shared neighbors ratio in relation to all types of the neighbors of the two vertices.

Density of vertex is a number of neighborhoods of the vertex. Where density is a vector of the number of adaptive neighbors for each element and density (a) is the number of the adaptive neighbors of the individual a.

MATERIALS AND METHODS

Our first contribution: Upgrade PIFS to use in recognition patterns where most of the researcher used a fractal dimension. PIFS applied on the single image while upgrade PIFS applied on multi images where range and domain blocks extract from all images and each range block machining with all domain blocks which extract from all images. The output of upgrade PIFS is not transformation parameters but the index of the image that has range block and the index of the image that has domain block. finally, count the number of occurs indexes to gather.

Second contribution: Develop graph clustering based on fractal features method can research on unweighted, undirected or weighted. Besides, there is no need for cluster numbers to be discovered. The methodology has two main stage. Firstly, ranges and domains matching technique is used to extract self-similarity features from the images. Secondly, using self-similarity features to use by graph clustering algorithm.

All images pass through two stages. The first known domain pool which is constituted by partitioning images into overlapping block fixed size and the second is ranges is formed from partitioning images into non-overlapping block fixed size. As far as the domain is concerned, the overlapping square results from image partitioning, the group D of domains involves all square degrees enriched by sides size 16. The domain amount doubles the range size. Consequently, it subsamples of 2×2 pixels to arrive a decreased domain by the identical sum of pixels like the range.

Regarding the range, the image gets partitioning to the form of squares of a non-overlapping that has sides size 8. Therefore, concerning every range, the algorithm attempts to detect a domain which provides the minimum error smaller. With respect to all recently invented ranges, the procedure is re-played, i.e., becoming suitable domains have been looked for ranges. The encoding terminations in.

Case there seems no ranges that still uncovered as well as save the image name. This in turn includes the block of the domain which making suitable the current range. RMSE is used to find the best fitting between range and domain (Fig. 3).

Self-similarity matrix is computed as result from matching operation where every entry represents the number of times matching between images. Algorithm 1 read k of images while the output is k×k matrix called Similarity Matrix M where entity of $M(t_1.t_2)$ represent the number of matching range from image t_1 with domain from image t_2.

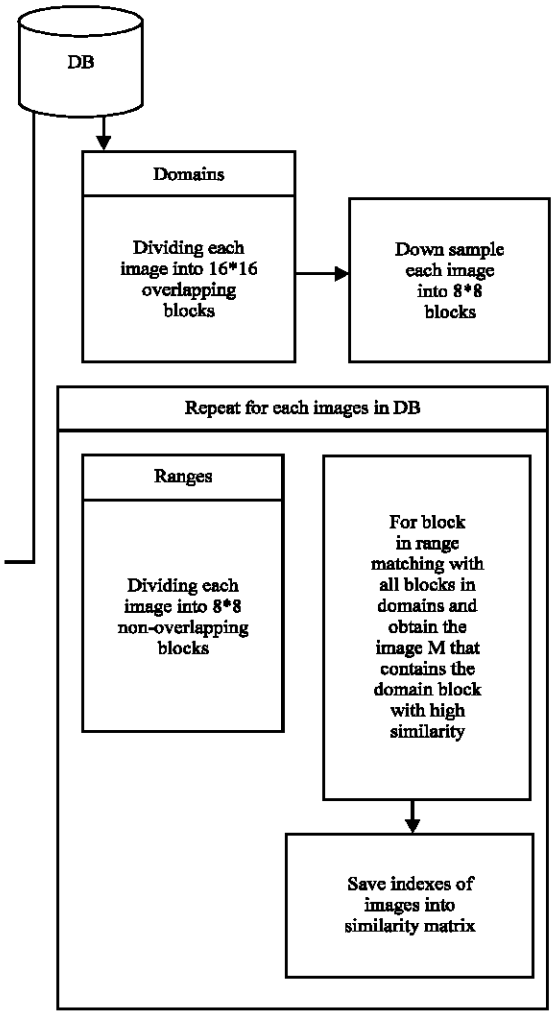

Fig. 3: Search operation to find the best matching between range and domains

Algorithm 1; Calculate similarity matrix among images:

Input: k face images
Output: M Similarity matrix of size k×k
Begin
/*Building domain Pool D*/
1: For each image I_i :I∈{1.2.3, ..., k}
2: Dividing I_i images into overlapping blocks D = {D_{i1}.D_{i2}…..D_{im}}.the size of each domain is 2b×2b then down sample to b×b size. Where m is number of domains in image and b is chosen value
3: End for
/*Matching range with domain Pool D*/
4: For each image I_i :I∈{1.2.3, ..., k}
5: Dividing I_i image into non-overlapping range blocks R= {R_{i1},R_{i2}, ... ,R_{in}} that size b×b Where n is number of ranges in image
6: End for
7: For each range R_{ij}∈R:i∈{1.2, ..., k} and j∈{1.2, ..., n}
8: For each domain D_{il}∈D:t∈{1.2, ..., k} and l∈{1.2, ..., m}
9: If RMSE(R_{ij}, D_{il}) is smallest M(i, t)= M(i,t)+1
10: End for
11: End for
12: Return M

The algorithm read k images as input to the algorithm. To build domain pool D in steps 1 and 2 all images are divided into overlapping blocks with size 2×2b because size of domain block must be double size of range block. Down sample must achieve on size of domain to facility the matching between ranges and domains with equal size. Average four pixels is used as down sample. Each domain indexed by image number which has this domain and domain number. Save all domains in pool domain D to using it later. With new round in step 3 take images again to dividing into non-overlapping blocks with size b×b to get ranges R. Each rang in R compare with all domains in domain pool to find high similarity domain then recorded i the index of image that contain range and t index of image that contain domain. The output of algorithm is M Similarity matrix of size k×k where k is number of input images. Rows in M represent i index of image that contain range and column represent t index of image that contain domain. The entity M (i.t) is number how many ranges in image i machining with domains in image t (Fig. 4). The output matrix is not symmetric.

The second algorithm used similarity matrix from the first stage as input. The suggested method can research on unweighted, undirected or weighted. Besides there is no need for cluster numbers to be discovered. Algorithm 2 schematizes the outline of the suggested method.

Algorithm 2; Graph clustering:

Input: Similarity Matrix W(N×N) for Graph G(V.E)
Output: Clustering C
Begin
1: For each vertex pair v_i .v_j ∈ V: i.j = 1 to N and I≠ j do
2: S[i.j] = SIM(v_i.v_j)Compute the similarity by Equation(1)
3: End for
4: For each vertex v_i in V where I = 1 to N do
5: Density(i) = the no of neighbors of vertex v_i
6: End for
7: Sort the items of the vector Density (V) in descending order
8: C(i) = 0 where i= 1, ...,N: c = 0
9: for each vertex v_i in Density (V) where i= 1, ...,N do
10: if C(i) is zero then
11: c = c+1 and C(i) = c
12: for each vertex v_j where vj ∈ (neighbors of vertex v_i) do if vertex if S[i,j] = arg max(S[j.k]) where k∈ (neighbors of vertex v_j)
13: if S[i.j] = arg max(S[j.k]) where k∈(neighbors of vertex v_j) then
14: C(j) = C(i)
15: End for
16: End for
17: Return C
18: End

No	1	2	3	4	5	6	7	8	9	10	11	12
1		5	23	4	18	2						
2	4		2	(33)	5	17						
3	36	11		5	7	4						
4	4	42	2		4	12						
5	21	13	10	18		1						
6	2	24	2	26	1			1				3
7		2	2	1				17	9	3	26	4
8					1		6		19	2	41	1
9			1				3	10		39	5	4
10			1				2	6	40		5	5
11							8	37	3	5		8
12				1	1	1		13	14	13	12	

Fig. 4: Shows what is entities of matrix M

The input of algorithm is similarity matrix of the undirected graph. In step 1 and 2 determine the adaptive neighbors for each node, the neighbors depend on the Jaccard similarity. The Jaccard similarity measures connectivity or the power of relationship among the pair of nodes. In step 5 compute the density of each node as follows.

Where density is a Vector (V) of the number of neighbors for each node and density (x) is the number of the neighbors of the individual x. Step 7 descending sorting the items of the density (V) vector. Steps 8-16. The first node in vector V must construct (create) the first cluster, since it has, the largest number of adaptive neighbors in density. All the adaptive neighbors of the first node in V must be located in this cluster with the condition it has the highest similarity with the first node. Therefore, the second node in V whose position corresponds the second element in density must be taken as candidate node. If (this candidate has been assigned to any existed cluster) then all its adaptive neighbors must be located in that cluster with condition it have the highest similarity with candidate node else this candidate will construct another new cluster and all its adaptive neighbors must locate in this new cluster with condition it have the highest similarity with candidate node. The process will continue until the last element in V has been clustered in its corresponding clusters.

Metrics of cluster quality: As far as cluster quantity is concerned it is normally categorized as a class of heavily related vertex which appears in connection with various sets in a certain graph. As such, lack of general as well as exact scientific cluster meaning is handed in the process of writing (Lancichinetti and Fortunato, 2009). On the other hand, assortments of different measurements which attempt to test the clustering quality are taken place via. catching the density of intra- cluster as well as sparsity of inter-cluster. Regarding G = (V, E) is an undirected graph in association with an adjacency matrix, three standards of measuring cluster quality are adopted in the current study, modularity, conductance and coverage. All of them are standardized in relation to the ultimate goal which scores range starting by 0 up to 1 where 1 represents the score that can be described as the ideal.

Modularity: Concerning modularity it compares the existence of every edge of intra-cluster of a certain graph with the edge probability that might be found in a haphazard graph (Newman and Girvan, 2004; Newman, 2004a, b). As a limit of resolution (Waltman and Eck, 2013) its algorithms of popular clustering functions objectively (Nawaz *et al*., 2015; Ren *et al*., 2015). Modularity is presented by Eq. 2:

$$\sum_{k}\left(e_{kk} - a_{k}^{2}\right) \tag{2}$$

where e_{kk}, stands for the intra-cluster probability of edges through cluster S_k whereas a_k, refers to the probability of one of two edges an intra-cluster within cluster S_k an inter-cluster incident in cluster S_k are:

$$e_{kk} = \frac{\left|\left\{(i,j): i \in s_k, j \in s_k, (i,j) \in E\right\}\right|}{\left|E\right|} \tag{3}$$

$$a_i k = \left|\left\{(i,j): i \in S_i k, (i,j) \in E\right\}\right| / \left|E\right| \tag{4}$$

and where $S_K \in V$.

Conductance: the cluster conductance can be identified via inter-cluster edges numbers to be divided through also the number of edges and an end point within the cluster but it can be also by the edges number which has not an end point within the cluster that appears lesser. The conductance a cluster is introduced in the form Eq. 5:

$$\phi\left(S_k\right) = \frac{\sum_{i \in sk,\, j \in s} A_{i,j}}{\min\left\{A\left(S_k\right), A\left(S_k\right)\right\}} \tag{5}$$

$$S_k \in V \text{ and } A\left(S_k\right) = \sum_{i \in s}\sum_{j \in v} Aij - \sum_{i \in s_k}\sum_{j \in s_k} A_{ij} \tag{6}$$

Eflects edge numbers in the endpoint within S_k. The graph conductance G is defined to as the conductance average for every cluster in relation to the graph, schematized from one. The graph conductance involves the range extends from (0-1) whereas the subtract has one the best score. Therefore, the graph conductance is presented in Eq. 7:

$$\phi(G) = 1 - \frac{1}{k}\sum_{k} j(S_k) \quad (7)$$

Coverag: Kobourov *et al.* (2014) involves the comparison of the division of intra-cluster edges of the graph with whole edges of the graph. Coverage is introduced as Eq. 8:

$$\frac{\sum_{i,j} A_{ij}\delta(S_i, S_j)}{\sum_{i,j} A_{i,j}} \quad (8)$$

where S_i refers to the cluster of the node i which is allocated whereas δ(a, b) represents 1 if a = b and 0 otherwise. It coverage, consists of the range of 0-1, since, 1 stands for that optimal score. On the other hand, coverage manages the concept of intra-cluster density, improving also greatly for the measure ends in a small clustering where in all nodes are allotted to the identical cluster.

The experimental data: The two database is used in this study. The first is the Feret database of images. These have been collected via. photographing 994 samples at different angles, during the course of 15 meetings that took place 1993 and 1996. The color Feret is mostly a color copy of the authentic Facial Recognition Technology (FERET) database that was discharged in 20001 and encompassed 14051 grayscale images. The database aimed at developing, examine and appreciate face recognition algorithms. The total images during the color FERET database amount 512 in the form of 768 pixels. Besides, the files located through PPM-format.

Second, faces of the ORL Database involves a group of face images gathered from April 1992 to April 1994 by the help of the lab. Ten dissimilar images for every of 40 discrete subjects are identified. Concerning particular subjects these images have gathered from various times, ranging from lighting, the expressions of face (open/closed eyes, smiling/not smiling) to the information of faces (no glasses/glasses). All of the images have considered in opposite to a dark analogous basis associating the subjects in a vertical, ahead position (with endurance for certain side action). Additionally, the image size is 92×112 pixels with 256 gray ranks per pixel.

RESULTS AND DISCUSSION

The results of the first part of proposed method (calculate similarity matrix among images) showed the images from same class are grouped together. In Fig. 5, the images from 1-10 which belong to one object have high, similarity. This study presents assessing how the proposed method performs. The B-cubed estimates the recall and precision for each item in a cluster on a certain data-set.

No	1	2	3	4	5	6	7	8	9	10	11	12
1		5	23	4	18	2						
2	4		2	33	5	17						
3	36	11		5	7	4						
4	4	42	2		4	12						
5	21	13	10	18		1						
6	2	24	2	26	1			1				3
7		2	2	1				17	9	3	26	4
8					1		6		19	2	41	1
9			1				3	10		39	5	4
10			1				2	6	40		5	5
11							8	37	3	5		8
12				1	1	1		13	14	13	12	

Fig. 5: Similarity matrix represent the similarity between images

The precision is the numbers of items in the same cluster belong to the same cluster. The recall returns the numbers of items of the same group are given to the same cluster.

Let O = {o_1, ..., o_n} be a set of objects and C is a cluster on O. Let L (o_i) (1 = i = n) be the class of o_i given by ground truth and C (o_i) be the cluster ID of o_i in C. Then, for objects, o_i and o_j, (1 = i,j = n, I-, j) correctness of the relative between tow objects (o_i and o_j) in same cluster C is assumed by Eq. 9: (Mon, 1986).

$$\begin{aligned} &\text{cor}(o_i o_j) = \\ &\{1\text{if } L(o_i) = L(o_i) \Leftrightarrow C(o_i) = C(o_j)\} \end{aligned} \quad (9)$$

B-cubed precision is given by Eq. 10:

$$\text{Prec} = \frac{\sum_{i}^{n} = 1 \dfrac{\sum_{oj \approx i \neq j, c(o_j)} \text{Cor}(o_{i,} o_j)}{\left\| \{O_j | i \neq j, C(O_i) = L(O_j)\} \right\|}}{n} \quad (10)$$

B-cubed recall is given by Eq. 11:

$$\text{Recall} \frac{\sum_{i}^{n} = 1 \dfrac{\sum_{oj \approx i \neq j, L(o_i) = (o_j)} \text{Cor}(O_i O_j)}{\left\| \{O_j | i \neq j, L(O_j) = L(O_j)\} \right\|}}{n} \quad (11)$$

The precision and recall are presented to assess how the proposed method feature extraction with k-mean the results, as shown in Table 1.

The results of the proposed method feature extraction with k-mean were compared with the results that were obtained from k-mean method with histogram feature

Table 1: Clustering accuracy

Database	No. of Images	Subjects	Precision (%)	Recal (%)
ORL	400	40	95	96
FERET1	700	75	90	92

Table 2: Comparison of the proposed method results with other method

Methods	Precision (%)	Recall (%)
k-mean with histogram	75	84
k-mean with the proposed method	95	96

Table 3: Clustering quality

Database	No.of images	Subjects	Modularity	Conductance	Coverage
ORL	400	40	0.90218	0.87341	0.858750
FERET1	700	75	0.87621	0.98211	0.775210
FERET2	1062	139	0.78523	0.98948	0.858190

extraction. The experimental results showed that our approach yielded 95% precision and 96% recall as shown in Table 2.

The results of the second part of proposed method. The clustering result is evaluated by modularity, conductance and coverage quality metrics the results show in Table 3. Our proposed method has quality metrics between 0.7 and 0.9 for all metrics which is acceptable for graphs clustering.

CONCLUSION

The study presents a fractal method to extract self-similarity features for graph clustering. This method yields the results of 95 and 96% for clustering precision and recall, respectively. In addition, the proposed method proved the possibility of the use fractal self-similarity features in image clustering without the use of fractal dimension. To sum up, the proposed method is very efficient for image clustering. It will be also useful for the clustering of other data.

As far as this research is concerned it has surveyed a sufficient strategy of graph clustering to partition the vertices depending upon connectivity between vertices. However, the more frequent strategy of connectivity is adopted to evaluate the relevance between vertices. In this regard, every cluster quality is concurrently estimated by coverage quality measures, modularity and conductance. The graph clustering yields the results of 0.90218, 0.87341 and 0.85875 for graph clustering quality measures modularity, conductance and coverage, respectively. However, the experiments findings in relation to the quality of the cluster. As such, the current notion appears suitable to distributed graph processing in relation to the partition of the whole graph of K subgraphs. Hence, the cluster numbers can be specified automatically.

REFERENCES

Andersen, R. and K.J. Lang, 2006. Communities from seed sets. Proceedings of the 15th International Conference on World Wide Web, May 23-26, 2006, ACM, Edinburgh, Scotland, ISBN:1-59593-323-9, pp: 223-232.

Everitt, B., S. Landau and M. Leese, 2001. Cluster Analysis. 4th Edn., Taylor & Francis, Milton Park, Didcot, UK.

Ino, H., M. Kudo and A. Nakamura, 2005. Partitioning of web graphs by community topology. Proceedings of the 14th International Conference on World Wide Web, May 10-14, 2005, ACM, Chiba, Japan, ISBN:1-59593-046-9, pp: 661-669.

Jaccard, P., 1901. [Comparative study of the floral distribution in a portion of the Alps and the Jura (In French)]. Bull. Soc. Vaudoise Sci. Nat., 37: 547-579.

Kaye, B.H., 2008. A Random Walk Through Fractal Dimensions. 2nd Edn., John Wiley & Sons, Hoboken, New Jersey, USA., ISBN:3-527-29078-8, pp: 136-426.

Kleinberg, J. and E. Tardos, 2002. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. J. ACM. JACM., 49: 616-639.

Kobourov, S.G., S. Pupyrev and P. Simonetto, 2014. Visualizing graphs as maps with contiguous regions. Eurographics Conf. Visual., 4: 1-5.

Lancichinetti, A. and S. Fortunato, 2009. Community detection algorithms: A comparative analysis. Phys. Rev. E, Vol. 80 10.1103/PhysRevE.80.056117

Leodkin, M.A., T.A. Lebedkina and A. Jacques, 2009. Multifractal Analysis of Unstable Plastic Flow. Nova Science Publishers, New York, Hauppauge, ISBN9781606924686,.

Liu, G., N. Chen, C. Ou, Y. Liao and Y. Yu, 2014. Image feature extraction based on multifractal theory. Proceeding of the International 2014 IEEE Workshop on Electronics Computer and Applications, May 8-9, 2014, IEEE , New Taipei, Taiwan, ISBN:978-1-4799-4565-8, pp: 1023-1026.

Liu, W. and Y. He, 2010. Fractal features and fuzzy clustering based tree leaf segmentation from images with the complex background. Proceeding of International Conference on Computer, Mechatronics, Control and Electronic Engineering, August 24-26, 2010, IEEE, New Taipei City, Taiwan, ISBN:978-1-4244-7957-3, pp: 317-321.

Maimon, O. and L. Rokach, 2005. Data Mining and Knowledge Discovery Handbook. Springer, Berlin, Germany, ISBN:13-978-0-387-24435-8, Pages: 1395.

Mandelbrot, B.B., 1982. The Fractal Geometry of Nature. 3rd Edn., Henry Holt and Company, New York, USA., ISBN: 9780716711865, Pages: 468.

Mon, K.K., 1986. Self-similarity and fractal dimension of a roughening interface by Monte Carlo simulations. Phys. Rev. Lett., 57: 866-1986.

Nawaz, W., K.U. Khan, Y.K. Lee and S. Lee, 2015. Intra graph clustering using collaborative similarity measure. Distrib. Parallel Databases, 33: 583-603.

Newman, M.E., 2004b. Detecting community structure in networks. Eur. Phys. J. B. Condens. Matter Complex Syst., 38: 321-330.

Newman, M.E.J. and M. Girvan, 2004a. Finding and evaluating community structure in networks. Phys. Rev. E-Stat., 69: 026113-026128.

Newman, M.E.J., 2004. Fast algorithm for detecting community structure in networks. Phys. Rev. E, Vol. 69. 10.1103/PhysRevE.69.066133.

Ren, W., G. Li and D. Tu, 2015. Graph clustering by congruency approximation. IET. Comput. Vision, 9: 841-849.

Schaeffer, S.E., 2007. Graph clustering. Comput. Sci. Rev., 1: 27-64.

Shi, J. and J. Malik, 2000. Normalized cuts and image segmentation. IEEE. Trans. Pattern Anal. Mach. Intell., 22: 888-905.

Shih, A.Z., 2014. An examination of color image clustering by using fractal signatures. Proceedings of the 2014 International Conference on Machine Learning and Cybernetics, July 13-16, 2014, IEEE, New Taipei City, Taiwan, ISBN:978-1-4799-4216-9, pp: 410-413.

Tasoulis, S.K., I. Maglogiannis and V.P. Plagianakos, 2014. Fractal analysis and fuzzy c-means clustering for quantification of fibrotic microscopy images. Artif. Intell. Rev., 42: 313-329.

Waltman, L. and V.N.J. Eck, 2013. A smart local moving algorithm for large-scale modularity-based community detection. Eur. Phys. J. B., 86: 1-14.

Wein, C.J. and I.F. Blake, 1996. On the performance of fractal compression with clustering. IEEE Trans. Image Process., 5: 522-526.

Xu, X., N. Yuruk, Z. Feng and T.A. Schweiger, 2007. Scan: A structural clustering algorithm for networks. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 12-15, 2007, ACM, San Jose, California, USA., ISBN:978-1-59593-609-7, pp: 824-833.