Acta Scientiarum

http://periodicos.uem.br/ojs
ISSN on-line: 1807-8664
Doi: 10.4025/actascitechnol.v46i1.62608

TECHNOLOGICAL INFORMATION

An integrated exploration of heat kernel invariant feature and manifolding technique for 3D object recognition system

Subramaniam Usha[1*], Murugesan Karthik[1], Marappan Jothibasu[2] and Vijayaragavan Gowtham[3]

[1]Department of Electrical and Electronics Engineering, Kongu Engineering College, Perundurai, 638060, Tamilnadu, India. [2]Department of Electronics and Communication Engineering, PSG Institute of Technology and Applied Research, Coimbatore, Tamilnadu, India. [3]Automation Team Lead, Billed Right Healthcare Solutions, Tamilnadu, India. *Author for correspondence: E-mail: sushamangal@gmail.com

ABSTRACT. Spectral Graph theory has been utilized frequently in the field of Computer Vision and Pattern Recognition to address challenges in the field of Image Segmentation and Image Classification. In the proposed method, for classification techniques, the associated graph's Eigen values and Eigen vectors of the adjacency matrix or Laplacian matrix created from the images are employed. The Laplacian spectrum and a graph's heat kernel are inextricably linked. Exponentiating the Laplacian eigensystem over time yields the heat kernel, which is the solution to the heat equations. In the proposed technique K-Nearest neighborhood and Delaunay triangulation techniques are used to generate a graph from the 3D model. The graph is then represented into Normalized Laplacian (NL) and Laplacian matrix (L). From each Normalized Laplacian and Laplacian matrix, the feature vectors like Heat Content Invariant and Laplacian Eigen values are extracted. Then, using all of the available clustering algorithms on datasets, the optimum feature vector for clustering is determined. For clustering various manifolding techniques are employed. In the suggested method, the graph heat kernel is constructed using industry-standard objects which are taken from the Engineering bench mark Data set.

Keywords: Graph clustering; laplacian matrix; delaunay triangulation; graph heat kernel; manifolding techniques; structural pattern recognition.

Received on February 21, 2022.
Accepted on April 19 2023.

Introduction

Graph representation

Graphs are structures with a long history in mathematics and have been used practically in various disciplines of science and engineering. A graph is a representation of a set of components and their pair wise relationships. The elements are known as nodes or vertices, while the connections are known as edges. The set G = (V, E) define Graph G, which considers graphs with self-loops. Also, graphs with many edges of the same orientation linking the same node pair are not taken into account. In the proposed method the graph is taken as an unweighted and undirected graph.

A graph (Raveaux, Adam, Héroux, & Trupi, 2011) can be represented using one of several popular matrices, as shown in Figure 1. A matrix representation may provide cost-effective storage because most graphs used in image processing are sparse, but each matrix representation can also be thought of as an operator that acts on functions associated with the nodes or edges of a graph. (Spielman, 2007; Blondel, Gajardo, Heymans, Senellart, & Van Dooren, 2004).

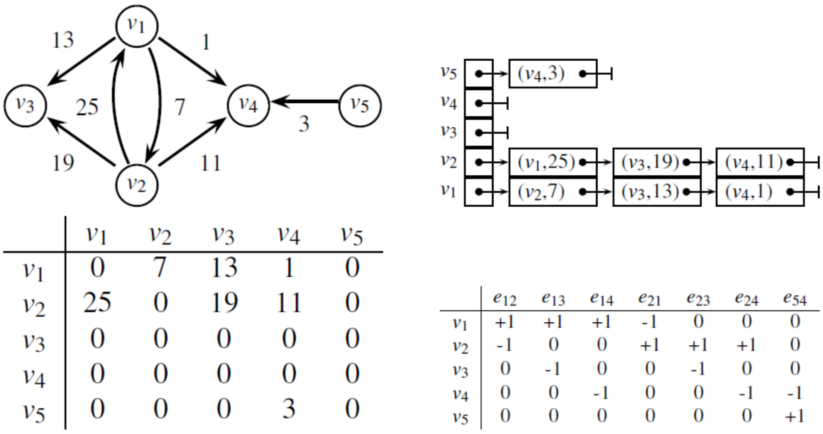
The adjacency matrix representation of graph G is shown in the Equation 1

$$\boldsymbol{A} = \begin{cases} 1, & if\ i = j \\ 0, & otherwise \end{cases} \tag{1}$$

The Laplacian matrix of an undirected graph G is shown in the Equation 2

$$\boldsymbol{L} = \begin{cases} di, & if\ i = j \\ -wij & if\ eij\ \in \varepsilon \\ 0, & otherwise \end{cases} \tag{2}$$

Adjacency lists have the advantage of using less memory than matrix representations. O (|V| x |E|) memory is required for the full incidence matrix, while O (|V|2) memory is required for full adjacency matrix. On the other hand sparse graphs, utilized for the purpose of significantly more economical storage. Figure 1 depicts weighted directed graph, adjacency list, and graph adjacency.


v_1
v_2
v_3
v_4
v_5
13
1
25
7
3
19
11
v_5 $(v_4,3)$
v_4
v_3
v_2 $(v_1,25)$ $(v_3,19)$ $(v_4,11)$
v_1 $(v_2,7)$ $(v_3,13)$ $(v_4,1)$

	v_1	v_2	v_3	v_4	v_5
v_1	0	7	13	1	0
v_2	25	0	19	11	0
v_3	0	0	0	0	0
v_4	0	0	0	0	0
v_5	0	0	0	3	0

	e_{12}	e_{13}	e_{14}	e_{21}	e_{23}	e_{24}	e_{54}
v_1	+1	+1	+1	-1	0	0	0
v_2	-1	0	0	+1	+1	+1	0
v_3	0	-1	0	0	-1	0	0
v_4	0	0	-1	0	0	-1	-1
v_5	0	0	0	0	0	0	+1

Figure 1. A Weighted directed graph, adjacency list, adjacency Graph.

Pattern matching (Fan, Medioni, & Nevatia, 1989) is the crucial function in high-level vision because it allows abstract graphical descriptions to be matched to one another. Many attempts have been made in the recent years to employ spectral graph theory in graph matching (Caetano, McAuley, Cheng, Le, & Smola, 2009; Abu-Aisheh, Raveaux, & Ramel, 2016) and point-set matching problems. One of the most crucial applications of spectral graph theory in computer vision is graph clustering, and another fascinating application is pattern recognition (Flickner, 1995; Schiele & Crowley, 2000; Vento, 2015) and routing, which can be used to streamline graph topologies. (Brun, Foggia, & Vento, 2020).

Literature Review

Spectral graph theory, can reduce a complex network to a lower dimensional state, making it easier to solve. The Fiedler vector is formed from the Laplacian eigenvalue of the second smallest eigenvalue and their related eigenvectors in that graph. This Fiedler vector holds the graph's maximal property and represents it in two-dimensional Euclidean space rather than other dimensions. As a result, these operate well in image segmentation and social networking processing, as well as any other complicated network. Wilson, Luo, and Hancock (2003) proposed, graph embedding method (Bunke & Riese, 2011) based on the graph's feature vector in vector space. The authors tested their technique on three different 3D polyhedral datasets of houses, each with a different view-based sequence. To construct graphs, corner detection and Delaunay triangulation techniques are utilised, and features of graphs such as leading eigenvalue, eigenmode volume, eigenmode perimeter, cheeger constant, and inter-mode of adjacency are discovered to form the matrix of each property. The objects are then classified using PCA, ICA, and MDS classification algorithms.

The structural graph representation of image objects was researched by Luo and Hancock (2001). To recognise the objects, the authors employed a mixture model of EM techniques and singular value decomposition. The research used these techniques to handle polyhedral 3D objects (Marini, Spagnuolo, & Falcidieno, 2007) with numerous view-based image sequences. Graphs are generated from images by applying a corner detection algorithm and then recovering the locations of those points as nodes. To generate the graphs and adjacency matrix the Delaunay triangulation process was utilized and the spectral graph theory is used to characterize graphs which are generated from the 3D polyhedral object. In their article the heat kernel graph (Kashima, Tsuda, & Inokuchi, 2004) is used for classification purpose. Corner detection and Delaunay triangulation methods are applied over an image to generate the graph. The matching Laplacian matrix of graphs is determined after the graph is formed. (Foggia, Percannella, Vento, 2014; Cordella, Foggia, Sansone, & Vento, 2004).

Inexact and exact graph matching (Bougleux, Gaüzère, & Brun, 2017; Bunke & Allermann, 1983) are the two major types of graph matching approaches introduced by Riesen, Jiang, and Bunke (2010). Because precise resemblance between the two graphs is required, exact matching approaches are computationally difficult; yet, in real-time applications, this is the key requirement. In inexact graph matching, a variety of techniques are used to match graphs, including ANN, Spectral methods, and Graph Kernels (Vishwanathan, Schraudolph, Kondor, & Borgwardt, 2010).

A method for learning graph matching was presented by Caetano et al. (2009). The training examples are pairs of graphs and the "labels" are matches between them. The authors used real image data to explore the experimental results, demonstrating that learning can increase the effectiveness of typical graph matching techniques. Linear assignment with such a learning method, in particular, appears to outperform state-of-the-art quadratic assignment relaxations. This conclusion implies that, for a variety of situations where the quadratic assignment was previously assumed to be required for acceptable results, linear assignment, which is significantly more efficient, may be sufficient if learning is done.

Spectral graph theory is used by Bai, Song, and Hall (2011) to characterise invariant picture structures. The authors proved that structure as a class identity for visual classification (Binford, 1971). They introduced graph energy as an effective method for extracting image structure.

Graph spectral approaches to the problem of point pattern matching (Conte, Foggia, Sansone, & Vento, 2004; Raveaux et al., 2011). were investigated by Tang, Shao, and Jones (2014) It focuses on how to employ graph spectral features (Batson, Spielman, Srivastava, & Teng, 2013) to successfully characterise point patterns in the presence of positional jitter and outliers. The attribute domain of feature points is represented by a set of 17 new local spectral descriptors (Xiao & Nelson, 2008) Weight graphs on nearby points are generated for each point in a given point collection, and then their normalised Laplacian matrices are produced. The distribution of the eigenvalues of these normalised Laplacian matrices is summed as a histogram to produce a descriptor based on the known spectral radius of the normalised Laplacian matrix. Finally, for retrieving correspondences between point-sets, the suggested spectral descriptor is merged with the estimated distance order.

A fully automated content-based image query system analysed the problem of matching point (Foggia, et al., 2014) sets over features taken from photographs. A novel solution to the problem proposed, is based on a combination of techniques from the literature. It provides a non-iterative technique for feature matching based on spectral methods, which combines a number of similarity metrics that quantify measures of correspondence between the two sets of features. The technique's versatility allows it to be easily applied in a variety of contexts, eliminating the domain-specific constraints of previous methodologies. The suggested technique is to test in a variety of case studies, including synthetic case studies, experimental biological data, and case studies based on well-known computer vision benchmarks.

A method for image recognition using an exact graph matching technique based on a genetic algorithm was suggested by Auwatanamongkol (2007). This method takes into consideration of differences between two graphs in terms of their shapes, numbers of nodes, and rotational radii. The proposed algorithm provides a high degree of accuracy in graph matching.

Numerous other techniques have also been proposed in the literature in addition to those of already mentioned. These techniques include probabilistic relaxation (Bengoetxea, Larranaga, Bloch, & Perchant, 2001); an Expectation-Maximization algorithm (Cross & Hancock 1998; Finch, Wilson, & Hancock, 1998), neural networks (Lee & Park, 2002), decision trees (Messmer & Bunke, 1999). However, some adjustments are needed to improve the efficiency and accuracy of graph matching, which is used in the image classification. It is also recommended to consider the entropy of the graph approach used for classification (Bai & Hancock, 2013).

3D Model Processing Using Graph

To generate the graph from an image or any 3D model, K-Nearest Neighbourhood (Kuncheva, 1995) graphs and Delaunay Triangulation logic are employed in the suggested method. The k-nearest neighbour graph (KNNG) is a directed graph in which each member is connected to its k closest neighbours. As a result, the KNNG is a graph G (V, E) given the element set V. As a result, in 3D CAD models, the object is represented by a set of vertices and directed edges. The proposed approach utilising the K-NN algorithm to determine the neighbourhood from the point clouds to group it. Each node's groups are represented in tree data structures. Therefore the 19 neighbourhood is used to build edges between the points. Figure 2 shows how the K-nearest neighbour graph is generated from the point cloud object.

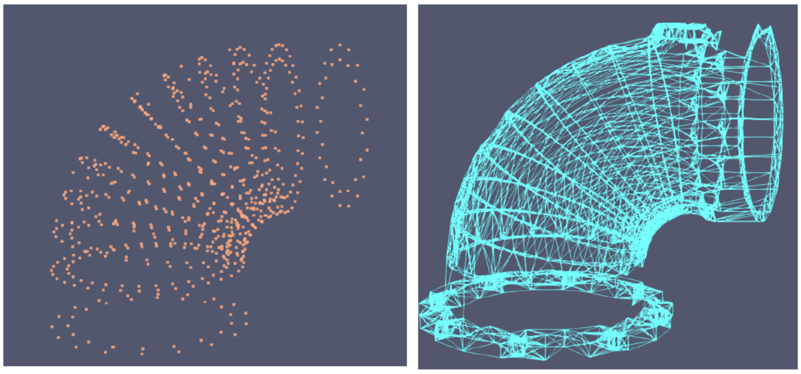


Figure 2. Appiied KNNG to Point Cloud.

After determining the location of critical points in an image, the node points are considered and these points are given to build the Delaunay triangulations. These points are extracted from the various feature detectors (Andreopoulos & Tsotsosb, 2013). As a result of the 1-Ring adjacency, the computational complexity of triangulation is reduced to O (n log n) instead of O(n4). Figure 3 depicts the use of 3D point cloud object data for Delaunay triangulation and also shows that the graph has 19638 nodes and 117558 edges.

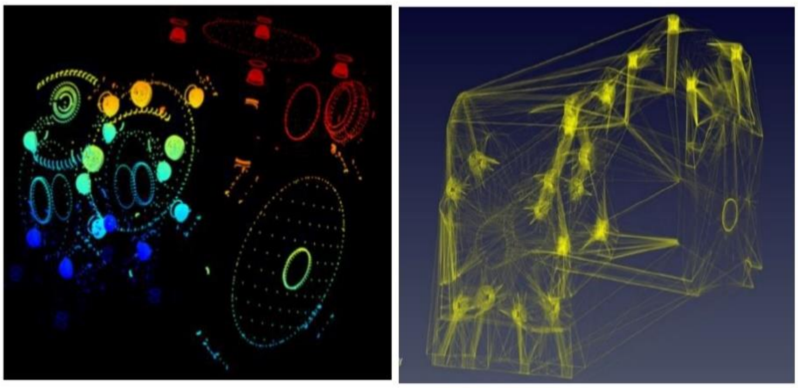


Figure 3. Delaunay Triangulation to Point Cloud.

Figure 4 depicts the steps that are involved in creating a graph from 3D image provided as input to the proposed Algorithm. The adjacency matrix of the image is created either using Delaunay Triangulation or the KNNG method from the 3D input models.

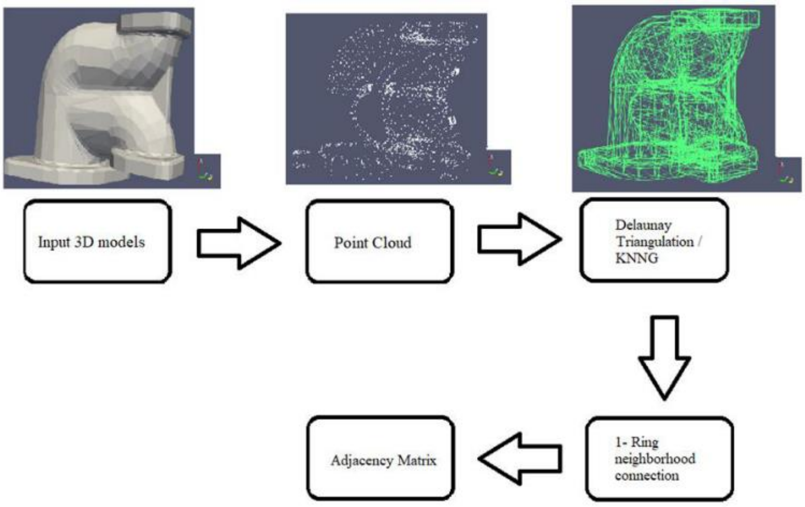
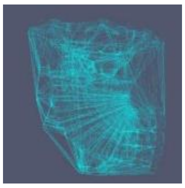
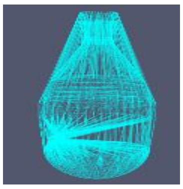
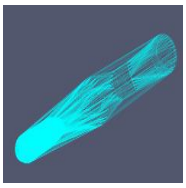
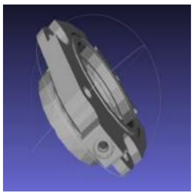
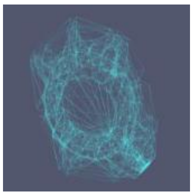
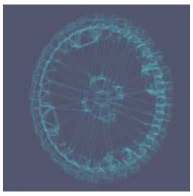

Input 3D models
Point Cloud
Delaunay Triangulation / KNNG
1- Ring neighborhood connection
Adjacency Matrix

Figure 4. Various steps in Graph formation.

Table 1 represents the 3D point objects and the graph generated using proposed algorithm. In the table the number of nodes, edges of the graph are generated and time in seconds required to generate the graph from the 3D object using the proposed algorithm is listed.

Table 1. 3D Point Cloud objects, Graph Generated and Properties.

S.No	Object	3D View	Graph Generated	Nodes	Edges	Time
1.	90 Elbow			3069	17116	0.158
2.	Collector			3202	18262	0.1797
3.	Ball Cutter			3427	23704	0.3185
4.	Oil Plane			1445	7525	0.0821
5.	Gear			1336	4879	0.2202

Heat kernel invariant

In the spectral graph theory, the heat kernel is very essential. The graph's heat kernel matrix encapsulates how information travels along the edges of the graph over time. A set of invariants like heat kernel trace, the zeta function, and heat content invariants can be computed using the graph's heat kernel matrix.

Heat kernel trace

Equation 3 shows the heat kernel trace, which is the sum of the diagonal elements of the graph's heat kernel matrix, where λi is the eigenvalue of the normalised Laplacian matrix.

$$Z(t) = Tr[h_t] = \sum_{i=1}^{|V|} \exp[-\lambda_i t] \tag{3}$$

Figure 5 depicts the heat kernel trace as a function of t for the various graphs (Xiao et al., 2005). As observed in the figure, the curves have a distinct shape that could serve as the basis for an appropriate representation to identify graphs. For example, the stronger the trace of the heat kernel at the origin, the more "hypercube" shaped graph will be the result. The size of $\lambda 2$ has an effect on the rate of decay of the trace over time, which is the measure of the degree of separation of the graph into tightly connected sub graphs (Bunke & Shearer, 1998) or clusters.

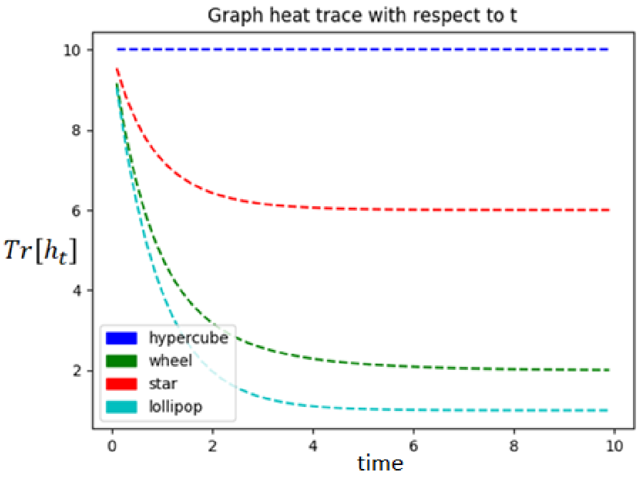


Figure 5. Heat Kernel Trace as a function of t

Equation 4 shows the extension of the heat kernel trace.

$$Z(t) = 1 + \exp[-\lambda_2 t] + \exp[-\lambda_3 t] + \cdots + \exp[-\lambda_N t] \tag{4}$$

Zeta function

The zeta function characterises graph structure by using the shape of the heat kernel trace function. The zeta function is coupled with the Laplacian eigenvalue and is strongly related to the heat kernel trace. In the Equation 5, the zeta function is defined.

$$\zeta(s) = \sum_{\lambda_i \neq 0} \lambda_i^{-s} \tag{5}$$

Heat Content Invariants (HCIs)

Because Zeta function offers a single scalar property for each value of time, the heat kernel trace is the only invariant useful for describing graphs. As a result, it must either be sampled throughout time or at a fixed time value. The heat content is calculated using Equation 6.

$$Q(t) = \sum_{u \epsilon V} \sum_{v \epsilon V} h_t(u, v) = \sum_{u \in V} \sum_{v \in V} \sum_{k=1}^{|V|} \exp[-\lambda_k t] \phi_k(u) \phi_k(v) \tag{6}$$

The heat content can be expanded as a polynomial in time and it is shown in Equation 7.

$$Q(t) = \sum_{m=0}^{\infty} q_m t^m \tag{7}$$

Because the co-efficient values are represented as a set of unique values for each graph, the Heat Content Kernel (Q_M) can be viewed as a set of invariants that can be used for graph characterization.

Results and discussion

This method will be applied to the Engineering Shape Benchmark (ESB) datasets and the heat invariants can be used for graph characterization. The dataset object is first separated into three structures based on the object's dimensionality. The second section demonstrates how to extract graph representations from a 3D point cloud. Third, using heat content invariant and symmetric polynomials, clustering has been done.

Database description and formulations

Engineering Models or Engineering Parts are having high genus, rounding features, and the existence of internal structure are used in the proposed method. Engineering models are waterproof, closed volumes that can represent pieces or assemblies. A wheel, for example, is a part, whereas a bike is an assembly. Part families and parametric models, which differ by relative dimensions of distinct local geometries, are popular in the engineering context. Engineering Shape Benchmark (ESB) is a database of engineering models with a well-established dataset. There are 867 parts and models in the neutral format in the ESB. Flat-thin walled, rectangular cubic prism and solid rotation are the three major super classes in the ESB classification. As a result, the ESB databases contain 45 classes.

There exists an issue while using this dataset as is because it is designed for meshes. Hence forth, dense point sets to represent engineering models are used in the proposed method. These model point sets were turned into a graph and used in the proposed algorithm. Here, ESB database models must be divided or grouped according to the complexity of the structures or the number of nodes in the models. As a result, the ESB was divided into three groups based on the number of nodes and structures.

First, Basic Structures (BST) which contains five classes of models, which are Prismatic stock, T Shaped, Bearing, Bolts, and Gears. This BST contain 25 objects and each class consists of five objects. BST objects based on the basic structure of models or model can be used to assemble the other parts. Second, Complex Structure (CST) contains ten classes of Engineering models, which are backdoor, pipe, bearing, cylindrical, intersection, clips, bearing block, elbow, handles, and 90-degree elbows. Contoured surface, handles, motor bodies, and flange are the third type of High Density Node (HDN). This HDN object is made up of extremely dense point sets. As a result, increased calculation overhead is required. Table 2 displays the list of Engineering model classes BST, CST, or HDN.

Table 2. List of Engineering Model Classes.

BST	CST	HDN
•Prismatic stock •T Shaped •Bearing •Bolts •Gear	•Backdoor •Pipe •Bearing •Cylindrical •Intersection •Clips •Bearing block •Elbow •Handles •90-degree elbows	•Contoured surface •Handles •Motor bodies •Flange

Figure 6 depicts the models selection from the established ESB datasets that have been grouped into specific classifications.

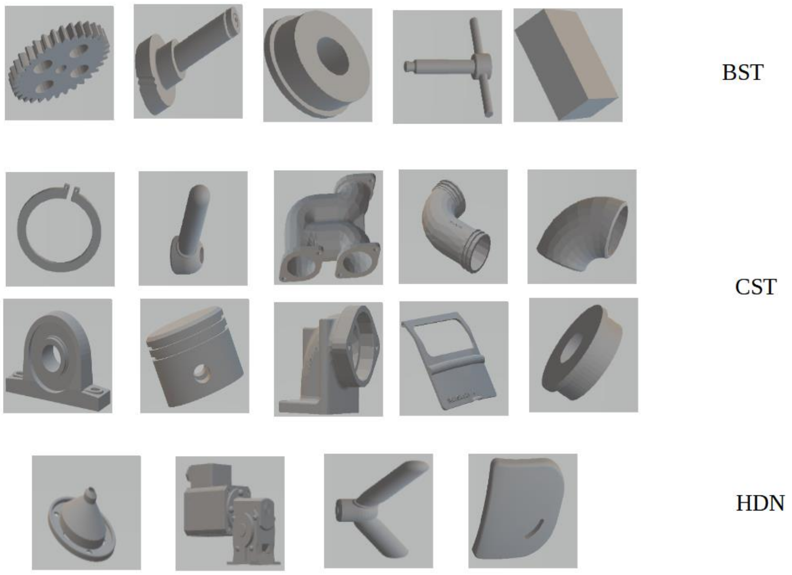


Figure 6. BST CST and HDN Models from ESP Dataset.

For the classification purposes, the proposed approach uses the BST dataset.

Heat Kernel Trace for ESB Models

The heat kernel trace is applied to the ESB datasets to verify if it can differentiate between different models. Engineering models are translated into graphs using Delaunay triangulation or KNN graph algorithm. The sum of the diagonal elements of the graph's heat kernel matrix, where λi is the eigenvalue of the normalised Laplacian matrix, transforms the model into a graph.

Figure 7 represents the heat kernel trace as a function of "t" for ESB models of the various ESB dataset images. As observed in the figure 7, the curves have a distinct shape that could serve as the basis for an appropriate representation to identify graphs. For instance, the more "Longer Driller" shaped the graph the more strongly peaked the trace of the heat kernel at the origin.

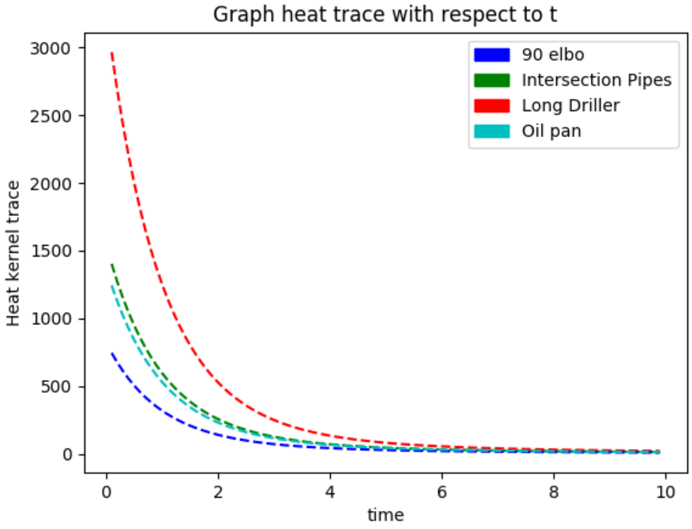


Figure 7. Heat kernel Trace of ESB.

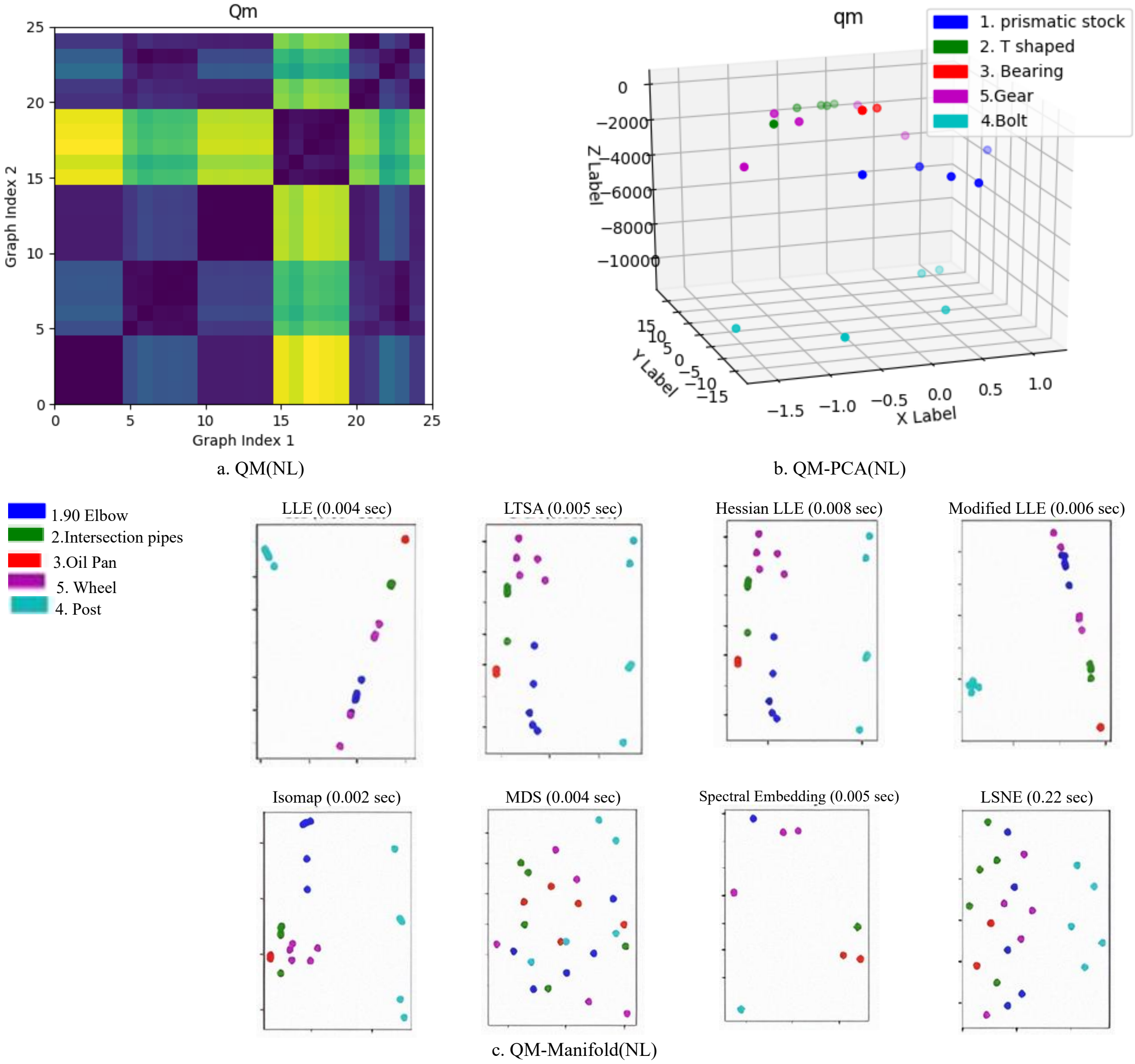
Graph clustering

All engineering models from BST are represented as graphs using Delaunay triangulation or KNN graphs utilising Normalised Laplacian (NL) and Laplacian matrix (L) in the proposed method. From each Normalized Laplacian and Laplacian matrix, the feature vectors like Heat Content Invariant $S = [\overrightarrow{B_1}|\overrightarrow{B_2}|\dots\overrightarrow{B_k}|\dots|\overrightarrow{B_M}|]$ and Laplacian Eigenvalues $B^{\rightarrow} = (l1, l2, \dots lk)T$are extracted. While, using all of the available clustering algorithms on BST datasets, determine the optimum feature vector for clustering. The Manifold learning techniques for clustering has been applied in the proposed method. They are Locally Linear Embedding (LLE), Local Tangent Space Alignment (LTSA), Hessian LLE, Modified LLE, Isometric Mapping (ISOMAP), Multi Dimensional Scaling (MDS), Spectral Embedding T Distributed Stochastic Neighbour Embedding (T-SNE).

Figure 8(a) shows the Euclidean distance between graphs in the BST datasets using Normalised Laplacian (NL) and Delaunay Triangulation. It also depicts the best pattern blocks for classifying each block. For instance, from 1-5 show a similar color and likewise, each 5-10, 11-15, 16-20, and 21-25 are showing the same color block. As a result, it is easier to determine which feature vector is best for the classification purposes. Figure 8(b) depicts the outcome of applying heat content kernels (Q_M) to PCA procedures. As seen in the figure, Q_M classifies the greatest number of models. The outcomes of various learning approaches like LLE, Hessian LLE, Modified LLE, ISOMAP, and Spectral Embedding are shown in Figure 8(c).

Figure 9(a) depicts the poor pattern block construction utilising Eigenvalues with Laplacian (L) and Delaunay triangulation that failed to distinguish the classes. The PCA of heat kernel content is displayed in Figure 9 (b). In comparison to the prior approaches, the PCA yields the poor outcome. The graph learning manifold results are displayed in Figure 9 (c). The input BST has no effect on the graph learning algorithms.

Figure 10(a) portrays the Euclidean distance between graphs heat content invariant values in the BST datasets using a KNN graph. From the result it is easier to determine which feature vector is the best for classification purposes. Figure 10(b) depicts the outcome of applying heat content kernels (Q_M) to PCA procedures. It is identified from the diagram AM is non-linear data; hence it is unable to classify the BST model. The outcomes of various learning approaches are shown in Figure 10(c). Hessian LLE, Modified LLE, ISOMAP and Spectral Embedding are the best clustering approaches for Q_M in manifolding techniques.

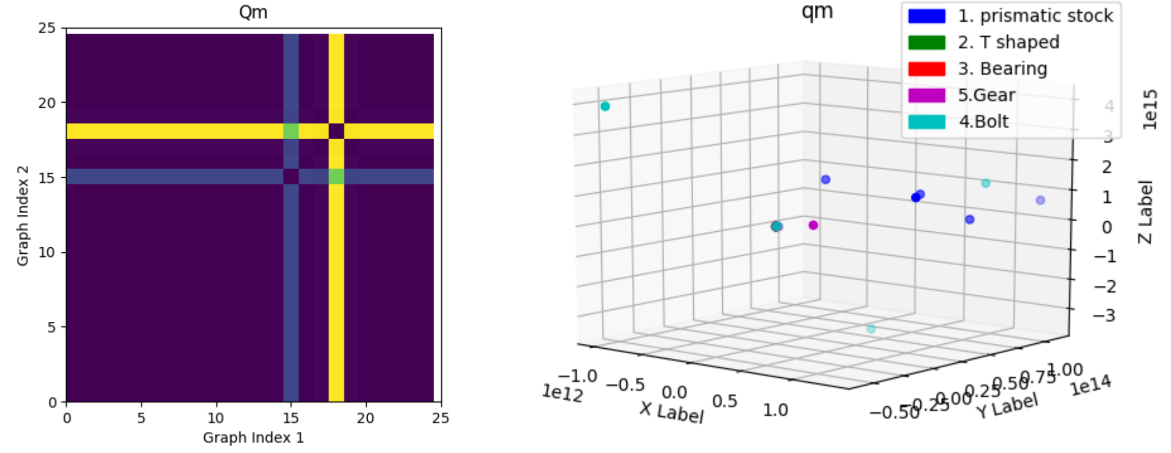


a. QM(NL)

b. QM-PCA(NL)

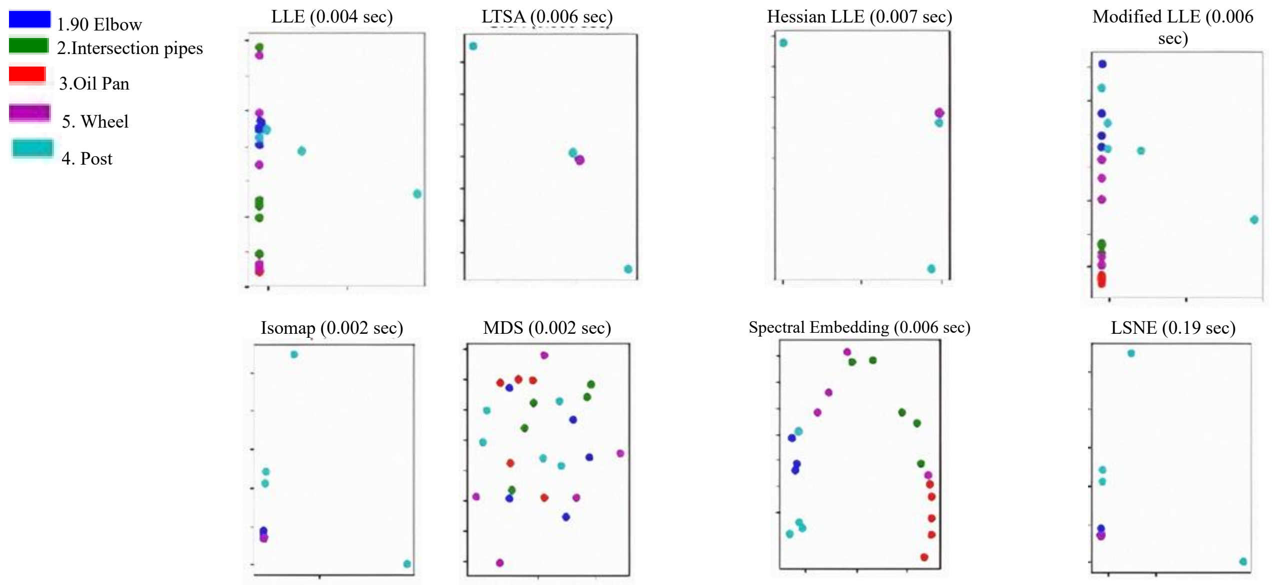
c. QM-Manifold(NL)

Figure 8. Q_M using NL and Delaunay Triangulation (BST).



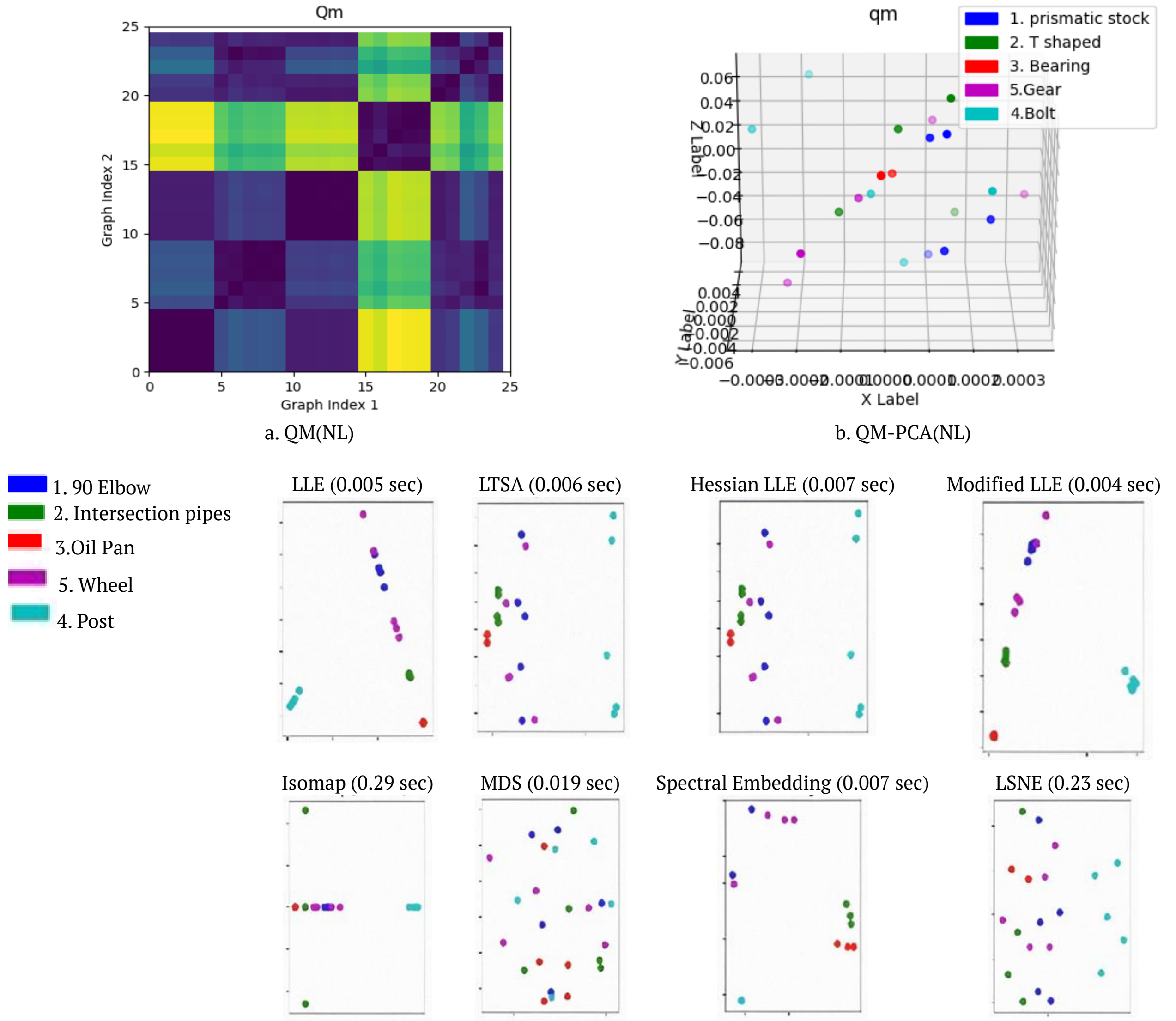
a. QM(L)

b. QM-PCA(L)



c. QM-Manifold (L)

Figure 9. Q_M of L and Delaunay Triangulation (BST).



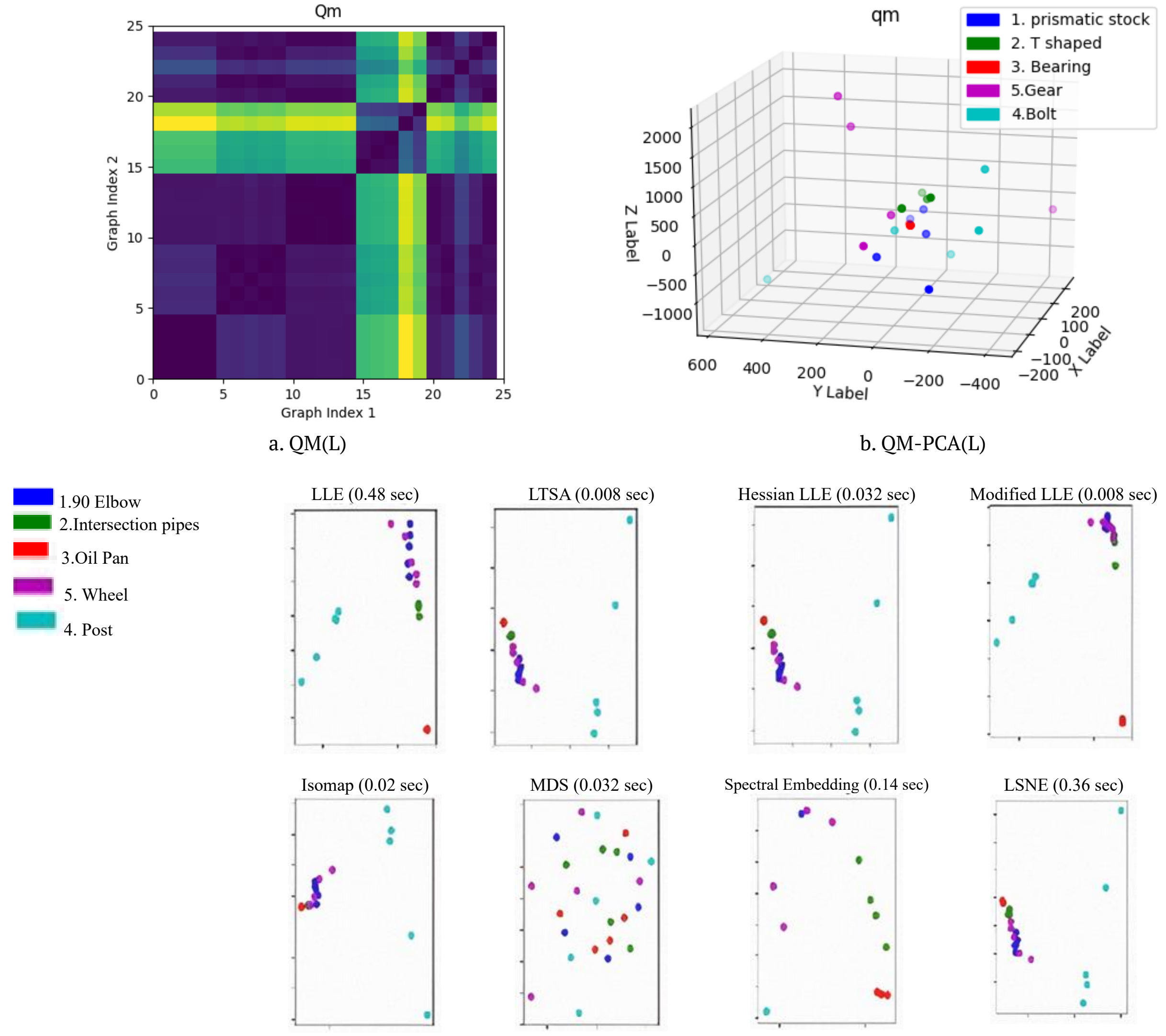
a. QM(NL)

b. QM-PCA(NL)

c. QM-Manifold(NL).

Figure 10. Q_M of NL and KNNG (BST).

In Figure 11(a) the Euclidean distance between graphs heat content invariant values in the BST datasets by KNN graph of Laplacian is demonstrated. Figure 11(b) shows the result of heat content kernel (Q_M) applied to the PCA techniques. Figure 11(c) shows the results of manifold learning techniques. In manifold techniques for Q_M gives best clustering results to LLE, Hessian LLE, Modified LLE, ISOMAP, and Spectral Embedding.



a. QM(L)

b. QM-PCA(L)

c. Q_M-Manifold(L).

Figure 11. Q_M of L and KNNG (BST).

The same method is employed using Eigen value feature and the results are tabulated. Table 3 shows the outcomes of various approaches on the BST datasets. According to Table 3, the model classification using KNN graph produces only average results when compared to the Delaunay triangulation. The results show that the Delaunay Triangulation methodology with Normalized Laplacian and Heat Content Invariant yields the best results. As a result, it is determined that Q_M is an invariant feature vector which produces good classification results with LLE, Modified LLE, Hessian LLE, ISOMAP, Spectral Embedding, and LTSA.

The CST dataset of the Engineering bench mark consists of 67 objects which include backdoor(7), pipes (15), bearing (8), cylindrical objects (7), intersection (9), clips (4), bearing block (4) ,elbow (5), handles (5) and 90 degree elbow(3).

The HDN dataset of the Engineering bench mark consists of 20 objects which include contoured surface, handles, motor bodies and flange.

The proposed method is applied to the CST data set and also HDN dataset for validation purpose.

Table 4 shows the outcomes of various manifolding techniques over CST and HDN datasets. The results show that the Delaunay Triangulation methodology with Normalized Laplacian and Heat Content Invariant yields the best results. As a result, it is determined that Q_M is an invariant feature vector that produces good classification results with LLE, LTSA and Hessian LLE manifold techniques.

Table 3. Comparative result of classification using delaunay triangulation and KNG Graph over BST ENG Data set.

Dataset	BST							
Techniques	Delaunay Triangulation (Proposed One)				KNN			
Methodology	NL		L		NL		L	
Features	Q_m	Eigen	Q_m	Eigen	Q_m	Eigen	Q_m	Eigen
PCA	22	20	7	15	7	23	18	13
LLE	25	19	7	20	22	23	23	20
LTSA	25	25	8	17	22	22	22	23
HESSIAN LLE	25	12	2	19	22	23	22	23
Modified LLE	23	13	10	19	23	19	21	20
ISOMAP	23	8	6	4	22	22	21	23
MDS	5	5	7	7	7	12	7	11
Spectral Embedding	25	16	19	18	23	23	23	21
T-SNE	14	4	12	6	17	17	22	6

Table 4. Comparative result of Classification using Delaunay Triangulation over CST dataset and HDN dataset.

Manifolding techniques	CST/67		HDN/20	
	Delaunay Triangulation Normalized Laplacian			
	QM	Eigen	QM	Eigen
PCA	37	32	19	13
LLE	59	37	20	13
LTSA	62	49	20	12
HESSIAN LLE	62	49	20	22
Modified LLE	61	51	17	11
ISOMAP	40	29	18	12
MDS	17	25	06	09
Spectral Embedding	42	40	14	07
T-SNE	40	11	05	05

Conclusion

The proposed approach makes an attempt at structural pattern recognition on a 3D point set engineering model. The suggested method integrates linear deformable model and heat kernel embedding to provide the generative model for graph structure. On ESB datasets, Delaunay triangulation produces the best graphs, and it can classify all objects using all classification methods using Normalised Laplacian feature vectors (QM), which outperform the Laplacian of graph in terms of results. The best classification outcomes on the BST, CST, and HDN datasets are found to be provided by Normalised Laplacian, LLE, M.LLE, LTSA, and HESSIAN LLE.

References

Andreopoulos, A., & Tsotsosb, J. K. (2013). 50 Years of object recognition: Directions forward. *Computer Vision and Image Understanding*, *117*(8), 827-891.

Abu-Aisheh, Z., Raveaux, R., & Ramel, J. (2016). Anytime graph matching. *Pattern Recognition Letters*, *84*, 215-224, DOI: https://doi.org/10.1016/j.patrec.2016.10.004

Auwatanamongkol, S. (2007). Inexact graph matching using a genetic algorithm for image recognition. *Pattern Recognition Letters*, *28*(12), 1428-1437. DOI: https://doi.org/10.1016/j.patrec.2007.02.013

Bai, L., & Hancock, E. R. (2013). Graph kernels from the jensen-shannon divergence. *Journal of Mathematical Imaging and Vision*, *47*, 60-69.

Bai, X., Song, Y.-Z., & Hall, P. (2011). Learning Invariant structure for object identification by using graph methods. *Computer Vision and Image Understanding*, *115*(7), 1023-1031. DOI: https://doi.org/10.1016/j.cviu.2010.12.016

Batson, J., Spielman, D. A., Srivastava, N., Teng, S-H. (2013). Spectral and Sparsification of graphs: Theory and Algorithms. *Communications of the ACM*, *56*(8), 87-94.

Binford, D. (1971). *Visual perception by computer*. Miami, FL: IEEE Conference on Systems and Control.

Blondel, V. D., Gajardo, A., Heymans, M., Senellart, P., & Van Dooren, P. (2004). A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM *Review*, *46*(4), 647-666.

Bougleux, S., Gaüzère, B., & Brun, L. (2017). *A hungarian algorithm for error-correcting graph matching*. In P. Foggia, C. L. Liu, M. Vento. (Eds.), Graph-Based Representations in Pattern Recognition. GbRPR 2017 (Lecture Notes in Computer Science, Vol. 10310, p. 118-127). Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-58961-9_11

Brun, L., Foggia, P., & Vento, M. (2020). Trends in graph-based representations for Pattern Recognition. *Pattern Recognition Letters*, *134*, 3-9. DOI: https://doi.org/10.1016/j.patrec.2018.03.016

Bunke, H., & Allermann, G. (1983). Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, *1*(4), 245-253.

Bunke, H., & Riese, K. (2011), Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition*, *44*(5), 1057-1067. DOI: https://doi.org/10.1016/j.patcog.2010.11.015

Bunke, H., &. Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letter*, *19*(3-4), 255-259.

Caetano, T. S., McAuley, J. J., Cheng, L., Le, Q. V., & Smola, A. J. (2009). Learn Graph Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(6), 1048-1058. DOI: https://doi.org/10.1109/TPAMI.2009.28

Conte, D., Foggia, P., Sansone, C., & Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *18*(3), 265-298.

Cordella, L., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Transacion on Pattern Analysis*, *26*(10), 1367-1372.

Cross, A. D., & Hancock, E. R. (1998). Graph matching with dual-step EM algorithm. *IEEE Transanction Pattern Analysis and Machine Intelligence*, *20*(11), 1236-1253.

Fan, T., Medioni, G., & Nevatia, R. (1989). Recognising 3-D Objects Using Surface Descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(11), 1140-1157. DOI: https://doi.org/10.1109/34.42853

Finch, A. M., Wilson, R. C., & Hancock, E. R. (1998). Symbolic Graph Matching with EM algorithm. *Pattern Recognition*, *31*(11), 1777-1790.

Flickner, M. (1995). Query by Image and Video Content: The QBIC System. *IEEE Transactions on Computer Science*, *28*(9), 23-32. DOI: https://doi.org/10.1109/2.410146

Foggia, P., Percannella, G., & Vento, M. (2014). Graph matching and learning in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *28*, 1450001.

Kashima, H., Tsuda, K., & Inokuchi, A. (2004). Kernel for graph. In *Kernel Methods in Computational Biology* (p. 155-170). MIT Press.

Kuncheva, L. I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters*, *16*(8), 809-814. DOI: https://doi.org/10.1016/0167-8655(95)00047-K

Lee, Y-L., Park, R-H. (2002). A surface-based approach to 3D object recognition using mean field annealing neural network. *Pattern Recognition*, *35*(2), 299-316. DOI: https://doi.org/10.1016/S0031-3203(01)00022-X

Luo, B., & Hancock, E. R. (2001). Structural graph matching using the em algorithm and singular value decomposition. *IEEE Pattern Analysis and Machine Intelligence*, *23*(10), 1120-1136. DOI: https://doi.org/10.1109/34.954602

Marini, S., Spagnuolo, M., & Falcidieno, B. (2007). Structural shape prototypes for the automatic classification of 3d objects. *IEEE Computer Graphics and Applications*, *27*(4), 28-37. DOI: https://doi.org/10.1109/mcg.2007.89

Messmer, B. T., & Bunke, H. (1999). A decision tree approach to graph and subgraph isomorphism detection. *Pattern recognition*, *32*(12), 1979-1998.

Riesen, K., Jiang, X., & Bunke, H. (2010). Exact and Inexact Graph Matching: Methodology and Applications. *Managing and Mining Graph Data*, *40*, 217-247.

Raveaux, R., Adam, S., Héroux, P., & Trupi, É. (2011). Learning Graph prototypes for Shape Recognition. *Computer Vision and Image Understanding*, *115*(7), 905-918. DOI: https://doi.org/10.1016/j.cviu.2010.12.015

Schiele, B., & Crowley. J. (2000). Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, *36*, 31-50.

Tang, J., Shao, L., & Jones, S. (2014). Point pattern matching based on line graph spectral context and descriptor embedding. *IEEE Winter Conference on Applications of Computer Vision*, 17-22. DOI: https://doi.org/10.1109/WACV.2014.6836123

Vento, M. (2015). A long trip in the charming world of graphs for Pattern Recognition Pattern Recogntion, *48*(2), 291-301. DOI: https://doi.org/10.1016/j.patcog.2014.01.002

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., & Borgwardt, K. (2010). Graph kernels. *Journal of Machine Learning Research*, *11*(40), 1201-1242

Wilson, R. C., Luo, B., & Hancock, E. R. (2003). Spectral embedding of graphs. *Pattern Recognition*, *36*(10), 2213-2223. DOI: https://doi.org/10.1016/S0031-3203(03)00084-0

Xiao, C. H., & Nelson, H. C. Y, (2008). Corner detector based on global and local curvature properties. *Optical Engineering*, *47*(5). DOI: https://doi.org/10.1117/1.2931681