Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xx.xxxx/ACCESS.2020.DOI

AGSF: Adaptive Graph Formulation and Hand-Crafted Graph Spectral Features for Shape Representation

BASHEER ALWAELY[1,2] and CHARITH ABHAYARATNE[2], (MEMBER, IEEE)

[1]Department of Computer Science, The Norwegian University of Science and Technology (NTNU),Trondheim, Norway (e-mail:basheer.alwaely@ntnu.no)

[2]Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, S1 3JD, United Kingdom (e-mail: c.abhayaratne@sheffield.ac.uk)

Corresponding author: Charith Abhayaratne (e-mail: c.abhayaratne@sheffield.ac.uk).

ABSTRACT Addressing intra-class variation in high similarity shapes is a challenging task in shape representation due to highly common local and global shape characteristics. Therefore, this paper proposes a new set of hand-crafted features for shape recognition by exploiting spectral features of the underlying graph adaptive connectivity formed by the shape characteristics. To achieve this, the paper proposes a new method for formulating an adaptively connected graph on the nodes of the shape outline. The adaptively connected graph is analysed in terms of its spectral bases followed by extracting hand-crafted adaptive graph spectral features (AGSF) to represent both global and local characteristics of the shape. Experimental evaluation using five 2D shape datasets and four challenging 3D shape datasets shows improvements with respect to the existing hand-crafted feature methods up to 9.14% for 2D shapes and up to 14.02% for 3D shapes. Also for 2D datasets, the proposed AGSF has outperformed the deep learning methods by 17.3%.

INDEX TERMS Shape recognition, graph signal processing, graph spectral theory, adaptive graph connectivity, hand-crafted features, graph spectral feature learning.

I. INTRODUCTION

OBJECT recognition in terms of shape analysis has recently received a great attention in the field of computer vision [1] and applications, such as, security [2], medical imaging [3] and human activity and pose understanding [4]. The detection of shape appearance, part-structure, occlusion, articulation, and local details play an important role in the ways of shape classification. Representation of these characteristics is particularly significant when it comes to distinguishing highly similar shapes. This is often the case in many existing shape data-sets which consist of similar and complex shapes leading to ambiguity in shape recognition [5]. For example, although, various shapes in FIGURE 1 can be easily distinguished by human vision, it is challenging for shape classification algorithms due to the similarity in global structures and indistinguishable local variations of these shapes. Thus, the capturing small local details and prominent parts as well as the global structure into shape models is an important factor in distinguishing between different objects. Further, this becomes even more difficult for 3D shapes due to the complexity and different view-points of shapes. This challenge has motivated us for this work to exploit the shape structure in terms of protrusions and fine details present within the global shape to propose a novel model for shape representation.

Previous work on shape classification include a wide range of methods such as graph matching [6], inner-distance [7], complex-network [8], short-cut [1], shape contexts [9], variable-dimensional local shape descriptors (VD-LSD) [10], point feature histogram (PFH) [11] and fast point feature histogram (FPFH) [12]. A comprehensive review of these methods can be found in [13] and [14]. These studies aimed to achieve the optimal representation of shapes by considering their outlines. However, main limitations of these approaches are high computational complexity, restrictions on some shape sizes [6] and sensitivity to noise [7]. Psychophysical and neuro-physiological studies have proposed a hypothesis for a structural representation of shapes in terms of object structures, parts and their positional relationships [15]. Further, studies on human vision have highlighted the importance of capturing the local details of the shape surface for the human visual perception of shapes [16]. More importantly, another study on human vision suggests that the visual cortex perceives and understands shapes by representing the shape boundary as a connected set of nodes [17], which has inspired us for our proposed method in this paper.

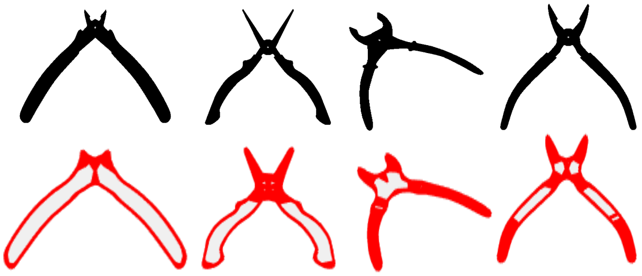


FIGURE 1: Challenging objects belonging to four different classes that have high conceptual similarity structures (top row) and their associated adaptively connected graph representation using the proposed method (bottom row). The new connections are highly concentrated at protrusions.

Inspired by the human vision literature, in this paper we propose a novel approach for shape representation by considering the shape as a connected graph, whose node connectivity is formulated adaptively, and analysing the spectral properties of the resulting graph. The proposed concept of adaptive formulation of connectivity, firstly computes a threshold to build a graph from shape nodes to capture complex shape structures and details. As shown in the bottom row of FIGURE 1, our method forms a graph with adaptive connectivity of nodes with connections (shown in red) highly concentrated at protrusions. Early stages and results of our proposed method [18] and a case study on application of it in 3D shape recognition [19] were presented as recent conference publications. In the present paper, we present a methodology representing both 2D and 3D shapes, by incorporating a new method to determine the threshold to build an adaptively connected graph and proposing additional features for classification. We also show rigorous experimental evaluation of the method using several 2D and 3D datasets leading to analysis and discussion of the recognition accuracy and system complexity including ablation studies. By 3D shapes, we refer to the shapes perceived by point clouds of 3D objects. The main contributions of this paper are:

1) Proposal of a novel graph-based representation of 2D and 3D shapes.
2) A new method for graph formulation with adaptive connectivity to represent shapes capturing their local and global characteristics.
3) Proposal of a new set of graph spectral features based on the node distribution of the adaptively connected graph for shape representation. Herein, we call them Adaptive Graph Spectral Features (AGSF).

The rest of the paper is structured as follows: Related works are reviewed in Section II. The proposed method is presented in detail in Section III followed by the performance evaluation in Section IV and concluding remarks in Section V.

II. RELATED WORK

Shape representation work in the literature can be categorized into five different groups: deep-learning methods, model-based methods, view-based methods, feature-based methods, and graph matching methods. Deep-learning-based methods [20]–[22] shown excellent performance in terms of recognition accuracy levels. The input features of these works are based on: exploitation of metric space distances (PointNet ++) [20], X-transformation (PointCNN) [21], and Kernel Point Convolution (KPConv) [22]. Deep learning methods often have high training times, high computational complexity and requires large training data. However, our focus in the present work is to propose a handcrafted features-based methodology, which does not require large training data.

Model-based methods include two types: shape-skeleton and shape-contour. The former constructs a tree model using object edges to form a shape descriptor, where the similarity measurement is based on tree matching approaches. For example, different methods are implemented by creating a shape descriptor prototype using short-cut [1], corresponding points [23], and skeleton pruning [24]. On the other hand, shape-contour models have relied on the boundaries of silhouette images, whose edges efficiently characterize the global structure of the object with a single closed curve provided there are no holes inside the object. An early study used Fourier descriptors to represent the shape [25], while the latest studies are based on convex details [26], circle view shape signature (CVS) with multi-views [27], and progressive shape-distribution-encoder [28]. In general, these approaches provide rich knowledge about shape structure in different strategies. However, the main issue in model-based methods is that the local details are omitted or completely neglected in their model. In other words, most model-based methods ignore the small protrusions or dense areas and focus on the global object structure instead. For each method, a specific dynamic program was used to identify the matching similarities between patterns. In some cases, even if the algorithm is not optimized, the matching program may increase the recognition score.

View-based methods measure the similarity between two objects based on different view angles. Particularly, they are based on analyzing and understanding the shapes rendered, distinguishing the models considering their visual similarity. Typical approaches based on view method includes multi-view depth line [29], and symmetric branch [30]. The main concern in these approaches is computing the view similarity for samples, which have different topology details as can be seen in the SHREC2010 dataset.

Feature-based methods involve extracting a set of features that provide an optimal representation of the shape structures followed by machine learning algorithm for classification processes. Feature-based matching typically requires more than one feature to describe a complex structure. Such features may include: a Scale Invariant Feature Transform (SIFT) [31], tree union [32], local phase [33], contour feature [34], distance [35], full shape [36], virtual retrieval [37], bag of words [38]. [10], variable-dimensional local shape descriptors (VD-LSD) [10], point feature histogram (PFH) [11], and fast point feature histogram (FPFH) [12]. The classification process is usually implemented using the Bayesian classifier [31], the Nearest Neighbors (NN) [33] or Support Vector Machines (SVM) [36].

Graphs and complex networks display useful topological features such as degree distribution, clustering coefficient, and hierarchical structure. Several methods have been proposed to explore the maximum probability of correspondence mapping between two patterns through the weight matrix based on the Eigen domain weighted graph matching [6], node degree [8], polynomial characterization [39], the center of clusters [40], spectral relaxation [41], higher order constraints [42] and Kronecker product of the graph adjacency matrices [43]. Recently several approaches have been proposed for large scale 3D point cloud representation. For example, the method in [44] presents a supervised 3D shape segmentation based on graph and deep learning. Similarly, the work in [45] proposes a deep learning approach-based on super point graph (SPG) for segmentation purposes. This method involves partitioning the 3D shape, down-sampling and embedding process followed by a graph convolution neural network to classify the shapes. However, both these methods aim to achieve an efficient cut of shapes using graphs, as opposed to modelling shapes using graphs as in our work.

In our recent work, we proposed graph spectral domain feature-based recognition for numbers [46]. The graph spectral feature-based methods that are primarily based on a fully connected graph formed by the shape were successful in recognizing global shape structures, but not on accurately representing local variations, as shown in FIGURE 1. To formulate the problem, we start with converting each shape into a set of nodes (for 2D by sampling the edges of silhouettes and for 3D by down-sampling the surface using Growing Neural Gas (GNG) [47]). The assumption is that the nodes at protrusions to have a high connectivity with each other compared to the nodes at the global shape boundary where less variations are present. The connectivity is defined as the number of nodes that one node is connected with. The existing graph node connectivity can be categorized into three types:

1) Full connectivity: when each node is connected to all other nodes in the graph. [46]. This type of connectivity provides an efficient characterization of the global outline of the shape.
2) Special connectivity: for specific applications, vertices have their own connectivity without the ability to change it. An example of this is a graph connecting major cities with a road network.
3) K-Nearest Neighbour: where each node is linked to the nearest K nodes, and each node has K connections [48].

As graph spectral theory uses the edge distribution for each node for converting graphs into their spectral domain, in order to provide an accurate spectral representation, accurate formulation of the connectivity for each node capturing the shape characteristics is important. Therefore, the main problem we aim to solve is how to determine the adaptive connectivity for each node in a shape. In this work, we propose to formulate an adaptively connected graph by forming conditional connectivity, where nodes are connected if a certain condition is satisfied as in Section III-C. Early stages and results of our proposed method [18] and a case study on use of it in 3D shape recognition [19] were reported as conference publications. In the current paper, we present a methodology representing both 2D and 3D shapes, by incorporating a new method to determine the threshold for conditional connectivity to build an adaptively connected graph and proposing additional spectral domain features for classification. We also show extended experimental evaluation of the method using several 2D and 3D datasets leading to analysis and discussion of the recognition accuracy and system complexity, including ablation studies.

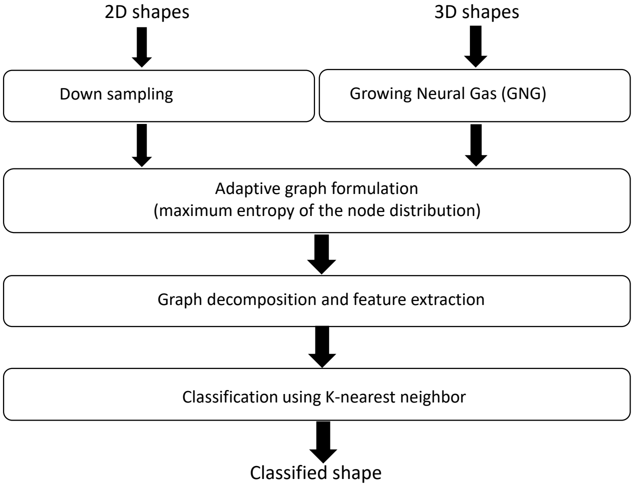

2D shapes
3D shapes
Down sampling
Growing Neural Gas (GNG)
Adaptive graph formulation
(maximum entropy of the node distribution)
Graph decomposition and feature extraction
Classification using K-nearest neighbor
Classified shape

FIGURE 2: Main steps of the proposed AGSF methodology.

III. THE PROPOSED METHOD

Our proposed method can be summarized into four steps as shown in FIGURE 2. The AGSF framework begins either by representing the shape as a collection of nodes or points. In the next step, these nodes are formulated into a graph with adaptive connectivity among nodes using the newly proposed adaptive graph formulation algorithm. This is followed by the spectral decomposition of the graph connectivity structure and feature extraction on graph spectral bases. Finally, these features are classified for recognizing the underlying shape.

A. SHAPE CONVERSION INTO NODES

2D shapes in datasets are usually given as silhouette or as an outline image. In this approach, we convert silhouettes to an outline, so that all input 2D shapes can be considered as a contour path. The resulting contour path, P, is usually a smooth curve with N number of pixels. To reduce the

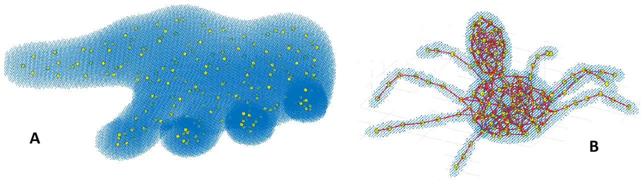


FIGURE 3: 3D shape representation using Growing Neural Gas.

complexity of the subsequent graph spectral decompositions, we choose n number of nodes, where $n < N$, to form a new down-sampled shape contour, $\hat{P}$, as follows:

$$\hat{P}(k) = P\left(\left\{\frac{Nk}{n}\right\}\right), \tag{1}$$

where $k = 0, 1, ..., n-1$ is the new node index and $\{.\}$ is the rounding to the nearest integer operator. $\hat{P}$ is then used as the nodes of the 2D shape, which will be later used for generating the graph with adaptive connectivity.

3D shapes are usually given as a point cloud with large number of points representing surfaces of the 3D shape. In order to make them manageable in the graph spectral decompositions, we down-sample the 3D shapes using the Growing Neural Gas (GNG) algorithm [47] due to its excellent quality, flexibility, and rapid adaptation to perform 3D surface representation of different objects. GNG is a simple unsupervised procedure to select the optimal pixels based on their distance, and it does not create any new pixels. The main characteristic of GNG is that the output neurons represent the topology of the shape with a fewer nodes. Although the GNG algorithm, outputs a connected graph, in this work we considered only coordinates of the nodes, ignoring the connectivity generated by GNG.

GNG starts with two nodes, randomly selected from a set of existing nodes. Then, it generates a signal based on the probability density between these nodes. After that, it finds the nearest node to both initial nodes. Based on Euclidean distance, the edges between these nodes will be updated based on the error function, which represents the difference in distance. These steps are repeated until the n nodes are selected. The use of GNG links is ineffective because it creates additional and unnecessary links outside the shape surface. For example, it creates links between different fingers, which are outside the geometry of the hand. Therefore, we cannot rely on GNG links. At the end of the training process, the GNG should satisfactory cover the shape regions as can be seen in FIGURE 3. Since GNG selects nodes regularly based on an unsupervised optimization process in such a way that these nodes have a uniform distribution inside the shape, the noisy pixels are removed by the GNG process. Node ordering is implemented from left to right for the 2D shapes, and bottom to top, and then left to right for the 3D shapes.

B. GRAPH SPECTRAL DECOMPOSITION PRELIMINARIES AND NOTATIONS.

We start with the preliminaries of graph spectral decomposition and our notation. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$ an undirected graph comprising $\mathcal{V}$, nodes or vertices, connected by $\mathcal{E}$, set of edges and represented by $\mathbf{A} \in \mathbb{R}^{n \times n}$, adjacency matrix comprising edge weights. We define the weight value, $\mathbf{A}_{i,j}$ corresponding to an edge, $e_{i,j}$ connecting vertices i and j is as follows:

$$\mathbf{A}_{i,j} = \frac{|e_{(i,j)}|}{\frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}|e_{(i,j)}|}, \tag{2}$$

which is the Euclidean distance $e_{(i,j)}$ between the vertices, i and j, normalised with the total lengths of edges in the graph averaged per node. We define the signal $\mathbf{r} : \mathcal{V} \to \mathbb{R}$, where the i^{th} component represents the Euclidean distance from the center (0,0,0) to the vertex i in $\mathcal{V}$. We also define, the node degree, $\mathbf{\Phi}_i$, as the number of edges incident on each node on $\mathcal{V}_i$. The combinatorial graph Laplacian matrix, $\mathbf{L}$, is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{3}$$

where $\mathbf{D}$ is the diagonal matrix of vertex degrees, whose diagonal components are computed as follows:

$$\mathbf{D}_{(i,i)} = \sum_{j=0}^{n-1} \mathbf{A}_{(i,j)}, \qquad i = 0, 1, ..., n-1. \tag{4}$$

Since, $\mathbf{L}$ is a symmetric positive semidefinite matrix, from spectral projection theorem, there exists a real unitary matrix, $\mathbf{U}$, that diagonalizes $\mathbf{L}$, such that $\mathbf{U}^t\mathbf{L}\mathbf{U} = \Lambda = diag\{\lambda_\ell\}$ is a non-negative diagonal matrix [49], leading to an eigenvalue decomposition of $\mathbf{L}$ matrix as follows:

$$\mathbf{L} = \mathbf{U}^t \Lambda \mathbf{U} = \sum_{\ell=0}^{n-1} \lambda_\ell \mathbf{u}_\ell \mathbf{u}_\ell^t, \tag{5}$$

where $\mathbf{u}_\ell$, the column vectors of $\mathbf{U}$, are the set of orthonormal eigenvectors of $\mathbf{L}$ with corresponding eigenvalues, $0 = \lambda_0 < \lambda_1 \leq \lambda_2 ... \leq \lambda_{n-1} = \lambda_{max}$ [50].

C. THE PROPOSED ADAPTIVELY CONNECTED GRAPH FORMULATION.

Now, we consider 2D and 3D shapes converted to a set of nodes (in Section III-A), which are the vertices, $\mathcal{V}$ in the graph formulation. The rest of this subsection will focus on how to generate the representative graph, i.e., to define $\mathcal{E}$ forming adaptive connections of vertices (in this subsection), followed by computing $\mathbf{A}$ for spectral decomposition and spectral domain feature extraction inSection III-D.

As discussed in Section II, although a fully connected graph provides an efficient representation of the global outline of the shape [46], the major drawback of this type is that local details are not reliably captured compared to the global outline. In addition, K-Nearest Neighbour connectivity does not reflect the topology structure, especially when using a

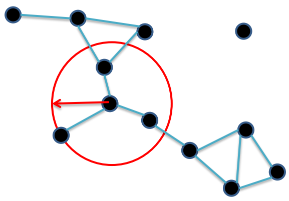


FIGURE 4: Conditional connectivity for a random graph showing vertices within the specified distance t are connected and others not connected.

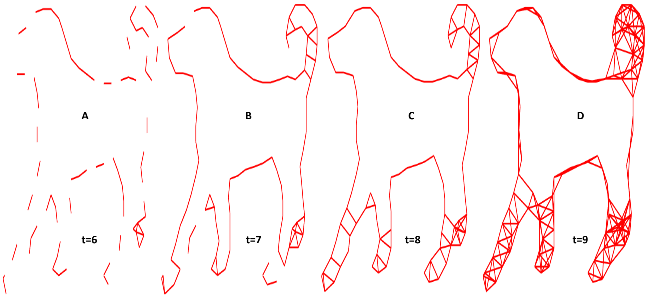


FIGURE 5: Graph construction using different threshold values showing how the graph connections grow.

small value of K, where nodes are only connected to their neighbours. Since we aim to classify more complex shapes, an appropriate connectivity is required to represent complex shapes within the Euclidean space. Thus, in this work, we propose adaptive connectivity, where vertices are connected if a certain condition is satisfied. In other words, $\mathcal{V}_i$ is connected to $\mathcal{V}_j$, if and only if they satisfy the condition, $|e_{i,j}| < t$, where t is a threshold distance. Using this connectivity type, there is no fixed number of connections at each vertex and the number of connected elements depends on the condition and the topology of the graph vertices. FIGURE 4 shows an example of how vertices are connected using a given threshold of distance between nodes.

The adaptive connectivity means that each vertex is connected to other vertices that fall in less than a certain dynamic distance. The dynamic threshold starts as a small distance value, then it steadily grows up till reaching a certain level to satisfy a certain conditions as explained in the rest of this section. This means that not all nodes are connected to each other, instead, the nodes that are close to each other tend to have strong connections. As a result, local details and the protrusions on the edge result in nodes with higher connectivity, which is captured by the Adjacency matrix and then by the spectral bases, where the spectral features are extracted.

As an example, FIGURE 5 shows generation of different graph connectivity using four different threshold values (t): 6, 7, 8 and 9 unit pixels to get graphs A, B, C, and D respectively. The corresponding graph eigenvalues (λ_ℓ) of $\mathbf{L}$ for the 4 graphs A, B, C, and D listed as vectors $\mathbf{V}_A$, $\mathbf{V}_B$, $\mathbf{V}_C$ and $\mathbf{V}_D$, respectively are shown below:

$$\mathbf{V}_A = [0,\ 0,\ 0,\ 0, ..., 0.50,\ 1.86, ..., 24.97],$$

$$\mathbf{V}_B = [0,\ 0,\ 0,\ 0,\ 0.01,\ 0.06, ..., 34.54],$$

$$\mathbf{V}_C = [0,\ 0.030,\ 0.050, ..., 49.94],$$

$$\mathbf{V}_D = [0,\ 0.031,\ 0.054, ..., 60.49].$$

Graph eigenvalues depend on the connectivity of the and thereby, reveal important characteristics of the underlying shape. Our method for choosing the threshold, t is formulated as below. To find the optimum t, we consider a range of thresholds, t_δ, as follows:

$$t_\delta = t_0(1+\delta), \tag{6}$$

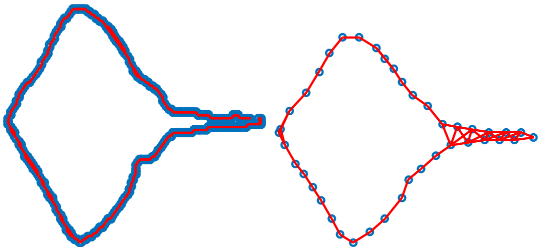
where $t_0 \in \mathbb{R}^+$ is the initial threshold for the minimally connected graph and $\delta \in \mathbb{N}$ is an increment distance with unit steps in the range $0 \leq \delta < (n-1)$. Here we have chosen n as the upper limit of δ. However, depending on the resolution of shapes in datasets we do not need to compute for all n as will be evident in the rest of this section. Since Eq. (6) reduces to $t_\delta = t_0 + t_0\delta$, it allows us to choose t_0 according to the sampling rates of data in different datasets and keep the integer increments for δ.

1) Finding t_0

The number of connected vertices for a graph with conditional connectivity depends on the threshold distance, t, and the topology of the graph vertices. This property of two graph structures can be compared by comparing their corresponding last eigenvalues, λ_{n-1}. The higher the λ_{n-1}, the higher the number of connections i.e., the connected nodes. Further, it can be observed that increasing t leads to increasing λ_{n-1}, as seen in the four examples in FIGURE 5 and their corresponding λ_{n-1}. Thus, we can summarise as $\lambda_{n-1} \propto t$.

For a given t value, there may be some instances where some nodes are not connected to any other node or only having one edge leaving gaps in the shape outline. In such cases, we can see clusters of connected vertices as in examples A and B in FIGURE 5. The number of clusters of connected vertices (ω) can be determined by counting the number of zero eigenvalues of the resulting graph structure for a given t [50]. As an example, the graph structure A in FIGURE 5 shows a dog with $\omega = 39$ clusters. Such a graph leads to $\lambda_0 = \cdots = \lambda_{38} = 0$ eigenvalues. Similarly, for B and $\mathbf{V}_B$ has $\omega = 4$ and $\lambda_0 = \cdots = \lambda_3 = 0$. For both C and D, it can be seen all nodes are connected as a single cluster with $\omega = 1$ and only one zero eigenvalue, $\lambda_0 = 0$. We can also see that more local details and protrusions are captured by means of edges when increasing the threshold values from $\mathbf{V}_A$ to $\mathbf{V}_D$.

We define t_0 as the minimum threshold distance that all vertices are connected as a single cluster i.e., $\omega = 1$. Initially, we start creating graphs over the shapes using a small distance to link the nodes. This usually creates unconnected



(a) $n = 400, t_0 = 2$. (b) $n = 40, t_0 = 12$.

FIGURE 6: Two examples of $\omega = 1$ for a large number of nodes $n = 400$ (left) and a small number of nodes $n = 40$ (right). Nodes are shown in blue and the graph connections in red.

graphs and $\omega > 1$ because of the small t. Therefore, t is gradually increased until the nodes are connected as a single group with $\omega = 1$. The minimum t to form such a minimally connected graph gives us t_0.

The value of t_0 is associated with the total number of nodes (n) in the sub sampled shape. The higher the n the smaller the value of t_0. Hence, we can summarise as

$$t_0 \propto \frac{1}{n}. \tag{7}$$

FIGURE 6 demonstrates the relationship between n and t_0 using the same shape with $n = 40$ and $n = 400$ nodes, respectively.

Note that we cannot define t_0 as the largest distance between any two nearest pair of nodes since the distance between the nodes in 3D shapes is not linearly related.

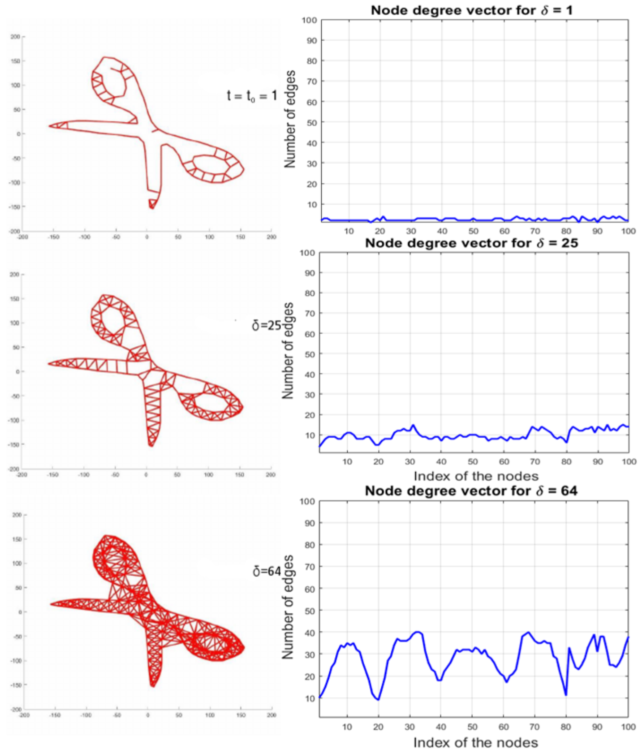


FIGURE 7: An example of three different levels of connectivity (in column 1) corresponding to various thresholds starting from t_0 to increasing δ values (from top to bottom rows) with their corresponding node degree $\mathbf{\Phi}^\delta$ plots (in column 2). A video showing growing graph connections and corresponding node degree vector evolution is available in the supplemental media materials.

2) Finding δ

For a given shape, the algorithm firstly finds t_0, the minimum threshold distance that all vertices are connected as a single cluster i.e., $\omega = 1$. We recall the graph node degree ($\mathbf{\Phi}$) is a vector comprising of number of connections (edges) for each vertex of the graph (shape outline). The node degree, $\mathbf{\Phi}^\delta$, for each t_δ is computed for each formulated graph connectivity corresponding to each δ. An example of graph edges and $\mathbf{\Phi}^\delta$ vectors for various δ values for a shape in the Tools dataset are shown in FIGURE 7. The first row in FIGURE 7 shows the graph corresponding to t_0 and $\mathbf{\Phi}^0$. The following rows show snapshots of the graph edges and the set of $\mathbf{\Phi}^\delta$ vectors up to the given δ. They show how the number of connections for each vertex grow with increasing δ. Now we need to analyse the shape of $\mathbf{\Phi}^\delta$ to determine the most suitable graph connectivity for representing the shape. For t_0 and for fully connected graphs, $\mathbf{\Phi}^\delta$ show less variation. As δ is increased, each node grows new connections resulting in $\mathbf{\Phi}^\delta$ vectors showing more variations as in FIGURE 7. As connections grow, local details and the protrusions of the shape, where more nodes are densely present, result in higher connectivity for nodes at those locations. As δ increases, $\mathbf{\Phi}^\delta$ vectors show growing number of connections making peaks corresponding to nodes at protrusions and local shape variations. For this, we analyse the amount disorder in $\mathbf{\Phi}^\delta$. Firstly, the normalised node degree, $\mathbf{E}^\delta$, is computed as follows:

$$\mathbf{E}^\delta = \frac{\mathbf{\Phi}^\delta}{\max(\mathbf{\Phi}^\delta)}. \tag{8}$$

This is followed by computing the metric we call weighted semi log normalised energy, $\mathbf{S}$, as follows:

$$\mathbf{S}_\delta = \sum_{i=0}^{n-1} \mathbf{E}_i^\delta \log_2 \left(\frac{1}{\mathbf{E}_i^\delta} \right). \tag{9}$$

The metric S is a measure of the degree of disorder of the normalised node degree vector, $\mathbf{E}^\delta$. This is inspired by the computation of entropy in information theory. We do not call it entropy as it does not involve probabilities. The S value increases when the variations in the vector are high and decreases when the variations are low. We choose the δ that corresponds to maximum $\mathbf{S}_\delta$ to determine the threshold T for graph formulation as follows:

$$T = t_0(1 + \underset{\delta}{\operatorname{argmax}}\, \mathbf{S}). \tag{10}$$

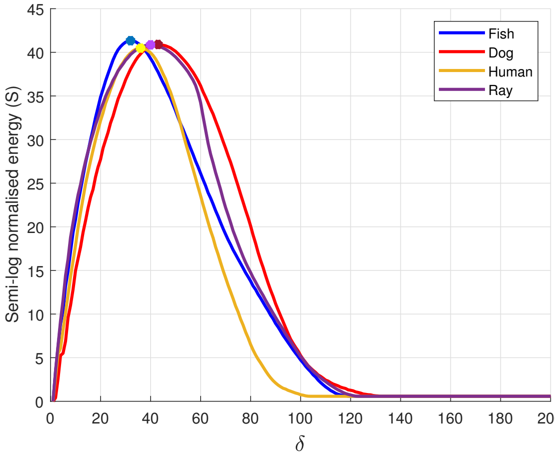


FIGURE 8: Examples of semi log normalised total energy, $\mathbf{S}_\delta$ curves with their maximum points (shown using •) for different δ values.

Examples of $\mathbf{S}_\delta$ for various δ values and their corresponding maximum points are shown for 4 shapes in FIGURE 8. The node degree vectors corresponding to maximum $\mathbf{S}_\delta$ and the resulting graph connections are shown in FIGURE 9. Our experiments show that despite the rotation angles of view, different shapes in the same class result in similar δ values producing the maximum $\mathbf{S}$ values. In this way, we find the adaptively connected graph for each shape. As a result, local details and the protrusions on the shape result in nodes with higher connectivity, which is captured by the Adjacency matrix and then by the spectral bases as shown in Section III-D. Also note that of the shapes are rotated, the distance between the nodes remains unchanged. Therefore, the connection between the nodes is not affected. As a result, $\mathbf{S}_\delta$ are not affected by rotation either.

D. PROPOSED ADAPTIVE GRAPH SPECTRAL FEATURE (AGSF) EXTRACTION

After each shape is converted to its adaptively connected graph representation, we propose the following features to extract from the spectral representation of the formulated graph structure. We select the rotation invariant features that can represent both global shape outline and local details. We propose a feature vector, $\mathbf{F}$, comprising of the following 3 components: $\mathbf{F}^1$, $\mathbf{F}^2$ and $\mathbf{F}^3$.

1) We capture the features from global outline of the shape by considering the distance vector $\mathbf{r}$, with $\mathbf{r}_i$ representing the distance to node i from the central point (0,0,0). Although $\mathbf{r}_i$ represents the global shape, in order to improve the discrimination among classes by considering the local variations, we modulate $\mathbf{r}$ with corresponding eigenvalues λ_i in $\mathbf{V}$ corresponding to the adaptively connected graph formulated in the previous section for the given shape sample, as follows:

$$\mathbf{F}_i^1 = \mathbf{r}_i\lambda_i, \qquad i = 0, \ldots, n-1. \tag{11}$$

Here, $\mathbf{r}$ represents spatial property of the shape, and λ corresponds to the spectral properties. Then, by modulating we aim to obtain spatial and spectral properties into the feature vector.

2) To compute $\mathbf{F}^2$, we consider the node degree, $\mathbf{\Phi}^\Delta$, of the adaptively connected graph for the shape sample, where $\Delta = \text{argmax}_\delta\, \mathbf{S}$. Since it corresponds to the adaptive connectivity graph formulation, this vector captures local details adequately. The normalised vector, $\mathbf{E}^\Delta$, is used as $\mathbf{F}^2$.

$$\mathbf{F}^2 = \mathbf{E}^\Delta. \tag{12}$$

3) The final feature component, $\mathbf{F}^3$, consists of a variety of other statistics of $\mathbf{\Phi}^\Delta$, the node degree corresponding to the adaptively connected graph for $\delta = \Delta$ as follows:

$$\mathbf{F}^3 = \left\{\mu(\mathbf{\Phi}^\Delta), \sigma^2(\mathbf{\Phi}^\Delta), ||\mathbf{\Phi}^\Delta||_2, \mathbf{S}_\Delta\right\}. \tag{13}$$

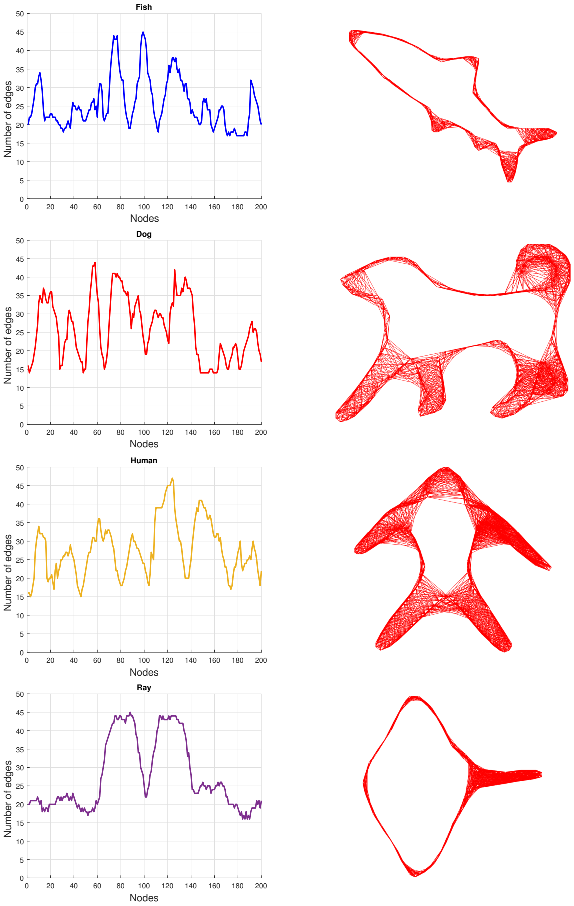


FIGURE 9: Examples of normalised node degree vectors, $\mathbf{E}^\delta$, (left column) for the δ corresponding to maximum $\mathbf{S}$ in FIGURE 8 and the resulting graph connections (right column).

These statistics (mean, variance, L_2 norm, and semi log normalised total energy) provide a set of features that is invariant to rotations.

The overall length of the feature vector, $\mathbf{F}$ is $2n + 4$, which is $\ll$ N.

E. MACHINE LEARNING

In this final step machine learning is used to learn the feature vectors generated in Section III-D for classification and recognition of the corresponding shape class. We have evaluated several classifiers including the Support Vector Machine with a cubic form as a kernel function (CSVM), the Nearest Neighbour (KNN), Classification Tree (CT), Discriminant Analysis (DA), Neural Network (NN). Based on several experiments conducted to select the optimal classifier, the Nearest Neighbour (KNN) with K= 1 shows the best performance compared to other classifiers in terms of accuracy and time processing, as shown in Section IV.

IV. PERFORMANCE EVALUATION

This section evaluates the performance of the proposed graph spectral features extracted from adaptively connected graph formulation from shapes on recognition of various 2D and 3D shapes with various sizes, orientations, articulation and scales from various datasets. All algorithms were implemented using MATLAB R2018a on a PC with Intel 3.6 GHz processor and 16 GB RAM.

A. DATASETS

The experiments were based on the following four 2D shape datasets and two 3D shape datasets:

1) ETU10 silhouette dataset is one of the most well-known 2D datasets [51]. This dataset has 10 classes with 72 shape samples in each class i.e., 720 total images. Each class contains instances of different views of the shape leading to at least 5-degree rotation difference for each instance. The ten classes: Bed, Bird, Fish, Guitar, Hammer, Horse, Sink, Teddy, Television and Toilet are numbered 1 to 10, respectively.
2) The tool dataset [63] is one of the most challenging 2D datasets, as it has a conceptual similarity within its shape classes. It consists of 35 articulated silhouette shapes, which are classified into four classes: 10 scissors, 15 pliers, 5 knives and 5 pincers respectively.
3) Kimia 99 dataset [64] consists of 9 classes with 11 shape samples in each class leading to 99 shapes. The nine classes, numbered from 1 to 9, in Tool dataset correspond to Fish, Hand, Human, Aeroplane, Ray, Rabbit, Misk, Spanner and Dog, respectively.
4) Kimia 216 dataset [65] consists of 18 classes with 12 2D silhouette samples in each class, i.e., 216 images. The 18 classes, numbered in 1 to 18, correspond to Bird, Bone, Brick, Camel, Car, Children, Classic, Elephant, Face, Fork, Fountain, Glass, Hammer, Heart, Key, Mink, Ray and Turtle respectively.
5) MPEG-7 CE-Shape-1 PartB (MP7-shape) dataset [66] consists of 70 classes with 20 samples in each class, i.e., 1400 2D silhouettes in total.
6) SHERC2010 dataset [67] consists of 10 point cloud object classes with 20 samples in each class leading to 200 3D objects in total. These samples are taken from McGill Articulated Shape Benchmark dataset. The classes, numbered from 1 to 10, include Ants, Crabs, Hands, Humans, Octopus, Pliers, Snakes, Spectacles, Spiders and Teddy respectively.
7) 3D shape benchmark dataset [68] consists of 19 classes with 20 samples per class totalling 380 3D shapes. Objects were presented in different orientations, scales and articulation, and that makes it one of the most challenging datasets. These classes, numbered 1 to 19, include Human, Cup, Glasses, Air plane, Ant, Chair, Octopus, Table, Teddy bear, Hand, Plier, Fish, Bird, Mech, Bust, Armadillo, Bearing, Vase and Four Leg respectively.
8) ModelNet10 dataset [61] consists of 10 classes of 3D shapes formed as a CAD model of the point could. The number of training and validation samples for each class vary from 106-889 and 50-100, respectively. These classes, numbered 1 to 10, include Bathtub, Bed, Chair, Desk, Dresser, Monitor, Night-stand, Sofa, Table and Toilet, respectively.
9) ModelNet40: [61] consists of 40 classes with the number of training and validation samples for each class vary from 64-889 and 20-100, respectively.

B. RECOGNITION PERFORMANCE OF THE PROPOSED METHOD - AGSF

TABLE 1 shows accuracy rates of the proposed method using different classifiers with 2D and 3D datasets using $n = 80$ and $n = 200$, respectively. All accuracy rates shown in this paper for the first 7 datasets use the average accuracy rates obtained from the k-fold cross validation scheme, using the k values shown in TABLE 1. Since ModelNet datasets provide separated training and testing samples, k-fold cross validation was not used for them. As can be seen, NN and KNN (with the normal Euclidean distance) classifiers result in the best accuracy rates among all classifiers. It must be also noted that KNN is faster than NN.

C. COMPARISON OF RECOGNITION RATES WITH THE EXISTING METHODS

TABLE 1 also compares with the performance of our previous work [18], [19], the existing hand-crafted features based methods and deep learning based methods. Our proposed hand-crafted features (AGSF) outperform the existing hand-crafted features based methods for 7 of the datasets. It also demonstrates that the use of the proposed energy function improves the performance compared to the variance in [19]. The confusion matrices showing recognition accuracy for each class in datasets that provide less than 100% overall accuracy

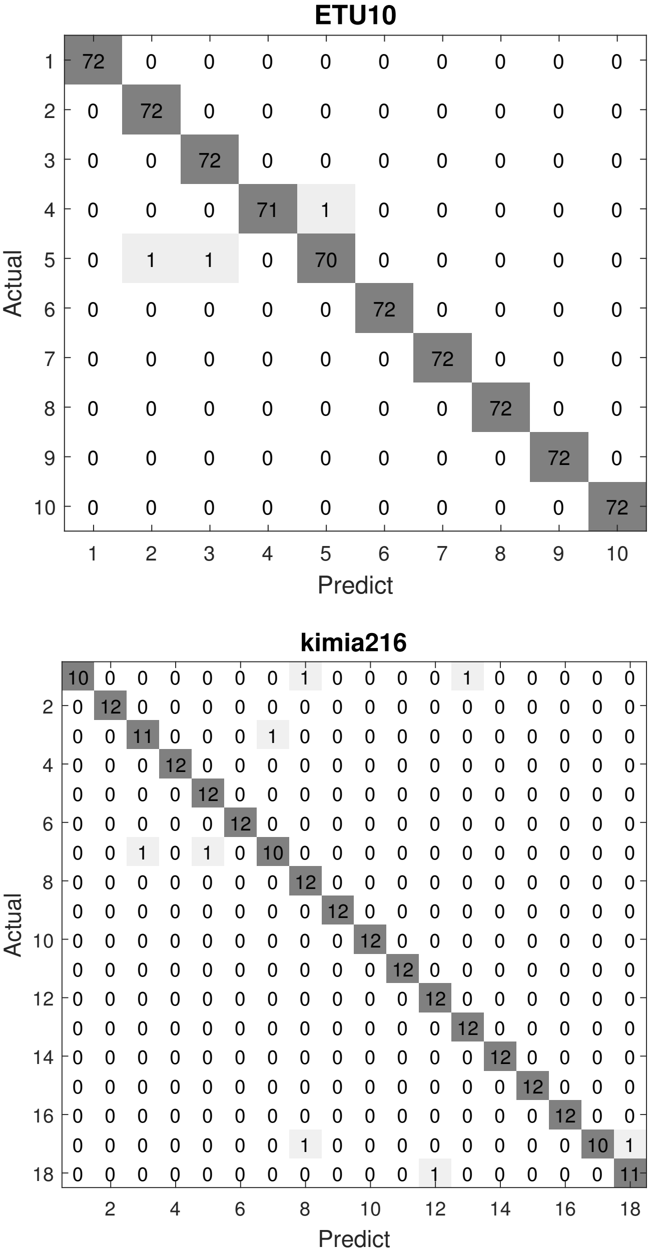
TABLE 1: Overall accuracy rates (%) for AGFS compared to the related work.

Dataset		Cross validation	Proposed AGSF + CSVM	Proposed AGSF + CT	Proposed AGSF + DA	Proposed AGSF + KNN	Proposed AGSF + NN	Our previous work	Existing hand-crafted features based methods	Existing deep learning based methods
2D	ETU10	10	99.58	94.18	89.54	99.58	**99.7**	99.31 [18]	97.50 [51]	N/A
	Tool	5	100	92.85	100	**100**	**100**	97.14 [18]	90.86 [34]	N/A
	Kimia99	9	97.98	90.82	91.91	**100**	99	96.00 [18]	100 [34]	N/A
	Kimia216	12	95.37	89.88	90.27	96.30	**96.8**	95.37 [18]	98.15 [32]	N/A
	MP7-shape	10	87.76	82.47	84.92	93.64	**97.4**	88.58 [18]	96.6 [52]	80.1 [53]
3D	SHREC2010	10	92.35	86.8	91.82	93	**94**	91.00 [19]	90.96 [54], 88.87 [55]	96 [56]
	3D benchmark	10	71.71	54.68	69.59	**76.32**	74.81	74.47 [19]	70.79 [8], 67.37 [57]	78.2 [58]
	ModelNet10	N/A	78.41	76	73.67	84.25	**87.11**	81.70 [19]	73.09 [59], 60.72 [60]	83.5 [61], 94.82 [62]
	ModelNet40	N/A	75.38	68.22	65.6	82.74	**86.43**	79.80 [19]	N/A	77 [61], 93.4 [62]

rates for 2D and 3D datasets are shown in FIGURE 10 and FIGURE 11, respectively.

Although, there is a significant similarity among the classes in ETU10 and Tool datasets, the proposed method manages to distinguish them well. For both Tool and Kimia99 datasets, shape samples are classified correctly with 100% accuracy. Although Tool dataset provides high similarity structures, the proposed method is found to be highly efficient in discriminating them as evident by the significant improvement in the accuracy rate. Similarly, despite having shape samples with different angles of views in ETU10 dataset, the proposed features outperform the existing methods with minimum confusion in recognising different classes. Kimia216 and MP7-shape are the most challenging 2D datasets due to having a small number of samples in each class compared to the total number of classes. For MP7-shape, the proposed hand-crafted AGSF even outperform deep learning [53].

For 3D datasets, the proposed hand-crafted features exceed performance of existing hand-crafted features based methods by 3.04% for SHERC2010 dataset, by 5.53% for the 3D shape benchmark dataset and by 14.02% for ModelNet10 dataset. As shown in FIGURE 11, the proposed method recognises all shape classes in SHREC2010 dataset with a high accuracy rate of 94%. The main confusing shape class is octopus (number 5), which is matched with spider class (number 9) due to its similarity graph structure leading to very similar eigenvalues and connected node distributions. Although, the 3D benchmark dataset is a very challenging dataset due to various angles of views, the proposed method has outperformed the existing hand-crafted features based methods. The ModelNet10 is a large dataset and usually used to evaluate deep learning methods. However, the proposed AGSF has achieved an overall recognition accuracy rate of 87.11% significantly outperforming the other hand-crafted features based methods. According to the confusion table, the most confusing class is the Night-stand (number 7) due to its structural similarity to Chair (number 3) and Desk (number 4). Also, a few errors appear in the Bathtub (number 1) class confused with the Dresser (number 5) class. For ModelNet40 our method achieves an overall accuracy rate of 86.43%. The accuracy of each class varies between different classes based on the number of training samples and the similarity among samples. As this dataset has mostly been used for evaluating

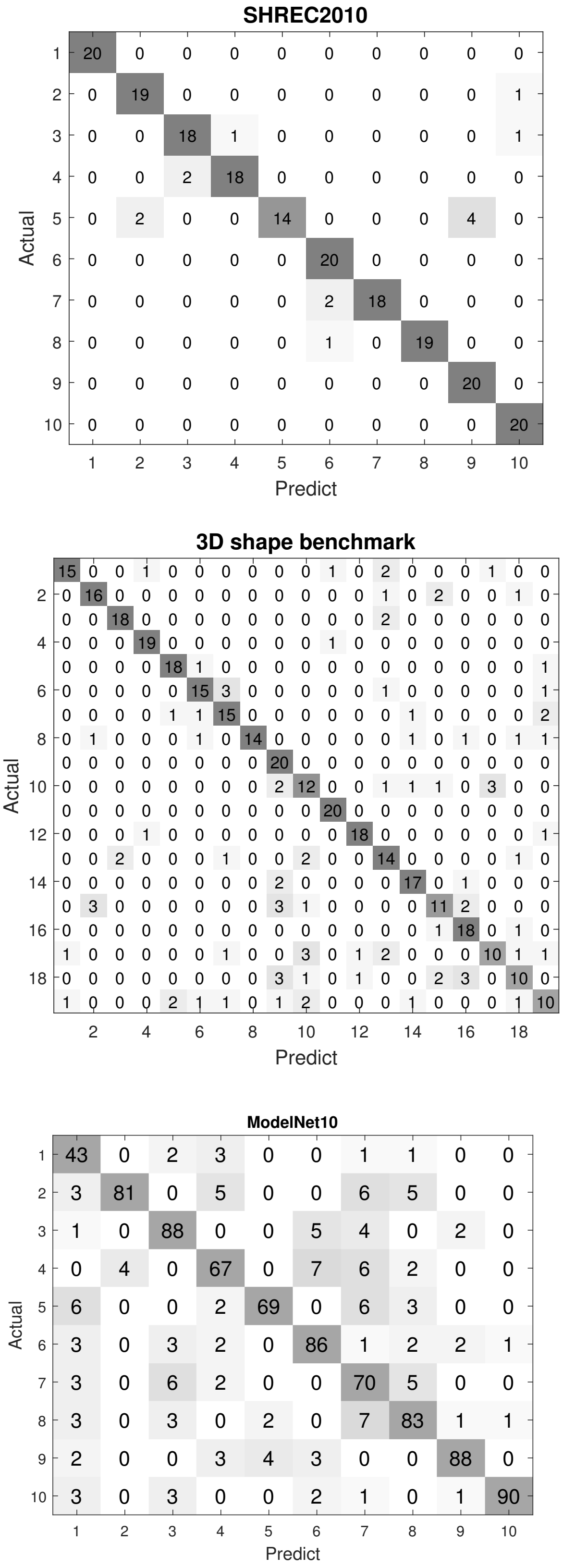


	1	2	3	4	5	6	7	8	9	10
1	72	0	0	0	0	0	0	0	0	0
2	0	72	0	0	0	0	0	0	0	0
3	0	0	72	0	0	0	0	0	0	0
4	0	0	0	71	1	0	0	0	0	0
5	0	1	1	0	70	0	0	0	0	0
6	0	0	0	0	0	72	0	0	0	0
7	0	0	0	0	0	0	72	0	0	0
8	0	0	0	0	0	0	0	72	0	0
9	0	0	0	0	0	0	0	0	72	0
10	0	0	0	0	0	0	0	0	0	72

		2		4		6		8		10		12		14		16		18
	10	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
2	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	11	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	1	0	10	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	10	1
18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	11

FIGURE 10: Confusion matrices of the 2D datasets, where the X and Y labels refer to the classes, which are described in Section IV-A.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3028696, IEEE Access



SHREC2010

	1	2	3	4	5	6	7	8	9	10
1	20	0	0	0	0	0	0	0	0	0
2	0	19	0	0	0	0	0	0	0	1
3	0	0	18	1	0	0	0	0	0	1
4	0	0	2	18	0	0	0	0	0	0
5	0	2	0	0	14	0	0	0	4	0
6	0	0	0	0	0	20	0	0	0	0
7	0	0	0	0	0	2	18	0	0	0
8	0	0	0	0	0	1	0	19	0	0
9	0	0	0	0	0	0	0	0	20	0
10	0	0	0	0	0	0	0	0	0	20

3D shape benchmark

		2		4		6		8		10		12		14		16		18	
	15	0	0	1	0	0	0	0	0	0	1	0	2	0	0	0	1	0	0
2	0	16	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	1	0
	0	0	18	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
4	0	0	0	19	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	0	0	0	0	18	1	0	0	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	0	0	15	3	0	0	0	0	0	1	0	0	0	0	0	1
	0	0	0	0	1	1	15	0	0	0	0	0	0	1	0	0	0	0	2
8	0	1	0	0	0	1	0	14	0	0	0	0	0	1	0	1	0	1	1
	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	2	12	0	0	1	1	1	0	3	0	0
	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
12	0	0	0	1	0	0	0	0	0	0	0	18	0	0	0	0	0	0	1
	0	0	2	0	0	0	1	0	0	2	0	0	14	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0	2	0	0	0	0	17	0	1	0	0	0
	0	3	0	0	0	0	0	0	3	1	0	0	0	0	11	2	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	18	0	1	0
	1	0	0	0	0	0	1	0	0	3	0	1	2	0	0	0	10	1	1
18	0	0	0	0	0	0	0	0	3	1	0	1	0	0	2	3	0	10	0
	1	0	0	0	2	1	1	0	1	2	0	0	0	1	0	0	0	1	10

ModelNet10

	1	2	3	4	5	6	7	8	9	10
1	43	0	2	3	0	0	1	1	0	0
2	3	81	0	5	0	0	6	5	0	0
3	1	0	88	0	0	5	4	0	2	0
4	0	4	0	67	0	7	6	2	0	0
5	6	0	0	2	69	0	6	3	0	0
6	3	0	3	2	0	86	1	2	2	1
7	3	0	6	2	0	0	70	5	0	0
8	3	0	3	0	2	0	7	83	1	1
9	2	0	0	3	4	3	0	0	88	0
10	3	0	3	0	0	2	1	0	1	90

FIGURE 11: Confusion matrices of the 3D datasets, where the X and Y labels refer to the classes, which are described in Section IV-A.

TABLE 2: Average accuracy rates in % for different k values k-fold cross validation. (The k value is shown in brackets next to the accuracy rate values.)

Dataset	Accuracy rate % **(k)**
ETU10	99.14 **(k=5), 99.56 (k=10), 99.56** **(k=20)**
Tool	100 **(k=5)**, 100 **(k=7)**
Kimia99	100 **(k=9)**, 100 **(k=11)**
Kimia 216	95.8 **(k=6)**, 96.3 **(k=12)**, 96.3 **(k=24)**
MPG7-shape	92.8 **(k=5)**, 93.64 **(k=10)**, 93.8 **(k=20)**
SHREC2010	92.5 **(k=5)**, 93 **(k=10)**, 94 **(k=20)**
3D benchmark	74.4 **(k=5)**, 76.32 **(k=10), 76.32 (k=20)**

deep learning based methods, no results for existing hand-crafted features-based methods have been reported in the literature.

Although our proposal is on hand-crafted features, we have included deep learning based methods for comparison. Out of all 2D shape datasets, used in our evaluation, deep learning results has been reported only for MPG7-shape dataset. Our proposed hand crafted AGSF has outperformed the deep learning based method by 17.3% for this dataset. However, for 3D datasets deep learning based methods have shown improvements of around 2% for smaller datasets and around 7% for large datasets.

D. COMPUTATIONAL COMPLEXITY.

The computational complexity of forming adaptive connectivity, graph spectral decomposition and feature extraction stages are $O(n^2)$, $O(n^2)$, and $O(2n^2)$, respectively. Then the overall complexity of the proposed method can be considered as $O(n^2)$ excluding the pre processing and classification steps. We also show the execution times of our method in TABLE 3. It includes the average time taken for the feature extraction, training and testing for 12 instances. In general, the average time taken to test a new sample is around 12 milliseconds, which reflects the real-time performance of our proposed method.

E. ABLATION STUDIES

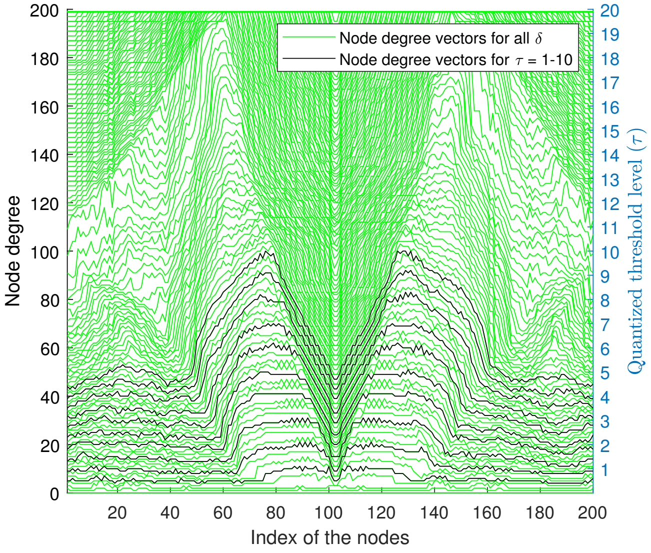
The effect of different parameters of out method including, k-fold cross validation, t, n, the adaptive connectivity and the proposed features are evaluated as follows:

1) k-fold cross validation

TABLE 2 shows the system performance based on different k-fold values for the datasets in FIGURE 10. In reporting the results, the best k for each dataset has been used.

2) The proposed threshold vs. number of nodes

In FIGURE 6, we showed that, for the same shape, the minimum threshold value (t_0) depends on the number of nodes (n). Therefore, we evaluate the proposed method using a wide range of n and t_δ values in order to evaluate the performance with respect to these parameters. In this exper-



yasseen2016shape

FIGURE 12: An example of node degree vectors for different(δ) values (in green) and quantised threshold levels (τ) (in brown) for $n = 200$.

TABLE 3: Minimum number of nodes at t_0 and the corresponding recognition accuracy rates and time taken.

Dataset	n	Accuracy rate (%)	Time (sec)
ETU10	55	99.56	7.47
Tool	45	100	2.84
Kimia 99	35	98.98	4.56
Kimia 216	65	95.83	6.13
MPG7-shape	40	92.4	10.47
SHREC2010	90	89	9.76
3D benchmark	130	71.7	11.63
ModelNet10	100	78.8	15.3
ModelNet40	100	77.7	37.42

iment, the accuracy rates for $30 \leq n \leq 200$ considering t_δ values quantised into τ levels were obtained. The quantised threshold level τ is defined as the threshold at which at least one node having a node degree of $5\tau\%$ for $\tau = 1, ..., 10$ with respect to n. FIGURE 12 shows an example of identification of τ levels for $n = 200$ with their corresponding t_δ and $\mathbf{\Phi}^\delta$. We also explore the optimal value of n that provides the highest discriminative representation between samples when $\delta = 0$. TABLE 3 shows that the range between 35-65 nodes is the efficient number to represent the shapes among all the datasets using only t_0.

3) Fully connected graph vs. adaptively connected graph

Next, we test the performance using fully connectivity graphs and the proposed conditional connectivity. TABLE 4 shows that the adaptively connected graph achieves better performance than that for the fully connected graph.

TABLE 4: Comparison of accuracy rates (%) for fully connected and the proposed AGSF.

Dataset	Fully connected + KNN	The proposed AGSF + KNN
ETU10	96.2	99.56
Tool	85.7	100
Kimia99	89	100
Kimia216	90.2	96.3
MPG7-shape	86.71	93.64
SHREC2010	88	93
3D benchmark	66.8	76.32
ModelNet10	76.3	84.25
ModelNet40	73.5	82.74

TABLE 5: The individual and combination accuracy rates (%) of the proposed AGSF + KNN.

Dataset	$\mathbf{F}^1$	$\mathbf{F}^2$	$\mathbf{F}^3$	$\mathbf{F}^1\mathbf{F}^2$	$\mathbf{F}^1\mathbf{F}^3$	$\mathbf{F}^2\mathbf{F}^3$	$\mathbf{F}^1\mathbf{F}^2\mathbf{F}^3$
ETU10	94.8	96.2	22.6	99.1	96.8	98.6	99.58
Tool	91.4	94.2	40	100	97.1	97.1	100
Kimia 99	90	93	26	98	93	95	100
Kimia 216	86.5	85.1	20.8	95.3	86.1	88.8	96.3
MPEG-7 CE-Shape-1	86.6	91.8	20.4	92.2	87.5	91.9	93.64
SHREC2010	89	90	28	92	90	92	93
3D benchmark	59.4	61.5	16.8	74.2	63.1	65.2	76.32
ModelNet10	78.74	80.5	21.5	83.81	80.28	80.94	84.25
ModelNet40	75.4	78	18.8	81.85	76.1	78.27	82.74

4) Elements of the feature vector

We also show a full ablation study of the proposed three kinds of hand-crafted features. TABLE 5 illustrates the importance of these features for shape representation. In this table, we can see that $\mathbf{F}^1$ and $\mathbf{F}^2$ have a greater impact on performance compared to $\mathbf{F}^3$.

F. DISCUSSION

TABLE 4 validates the contribution of the adaptively generated graph in shape representation. For, the large and challenging 2D silhouette dataset, MPG7, the proposed hand crafted features have outperformed the deep learning-based methods by 17.3%. discussed. Our proposed hand crafted AGSF has outperformed the deep learning based method by 17.3% for this dataset. As discussed in Section IV-C, deep learning methods have shown some advantage over hand-crafted features by around 2% and 7% for small and large datasets, respectively. It can be noted from Section IV-D and TABLE 3, that our approach has low complexity compared to the deep learning-based approaches and more suitable for smaller datasets in that case. Also note that the proposed hand-crafted features consider only shape features while the deep learning based methods learn both shape and texture features.

A few aspects need to be considered when choosing n in the initial shape sampling for 2D and further downsampling by GNG for 3D. Sampling should be high enough to capture the whole shape. For example, some samples in the ModelNet10 dataset show close similarity with some shapes in other classes after downsampling. Although GNG is useful as a pre-processing tool, it has several parameters, which have a direct effect on the quality of the representation, such as, the required output of n, number of iterations, and the average distance between nodes. At the GNG step, it is crucial to carefully select them to give reasonable points cloud to apply out method.

V. CONCLUSIONS

This paper has proposed a new set of hand-crafted features (AGSF) for shape recognition by exploiting spectral features of the underlying graph adaptive connectivity formed by the shape characteristics. To achieve this, we have proposed a new method for formulating an adaptively connected graph to represent shapes with an unique graph structure. This is followed by proposing graph spectral features to capture both global and local characteristics of the shape to train a classifier. The effectiveness of the proposed AGSF is verified by experiments on five 2D shape datasets and four 3D shape datasets. The proposed AGSF has outperformed the existing hand-crafted feature methods up to 9.14% for 2D shapes and up to 14.02% for 3D shapes. Also for 2D datasets, the proposed AGSF has outperformed the deep learning methods by 17.3%.

ACKNOWLEDGMENT

This work was carried out with the support of ERCIM 'Alain Bensoussan' Fellowship Programme.

REFERENCES

[1] L. Luo, C. Shen, X. Liu, and C. Zhang, "A computational model of the short-cut rule for 2D shape decomposition," in IEEE Transactions on Image Processing, vol. 24, no. 1, pp. 273–283, 2015.

[2] S. Chang, L. Chen, Y. Chung, and S. Chen, "Automatic license plate recognition," in IEEE Transactions on intelligent transportation systems, vol. 5, no. 1, pp. 42–53, 2004.

[3] L. Shen, R. Rangayyan, and J. Desautels, "Application of shape analysis to mammographic calcifications," in IEEE Transactions on medical imaging, vol. 13, no. 2, pp. 263–274, 1994.

[4] H. Liu, Q. He, and M. Liu, "Human action recognition using adaptive hierarchical depth motion maps and gabor filter," in Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 1432–1436.

[5] R. Basri, L. Costa, D. Geiger, and D. Jacobs, "Determining the similarity of deformable shapes," Vision Research, vol. 38, no. 15-16, pp. 2365–2385, 1998.

[6] S. Umeyama, "An eigendecomposition approach to weighted graph matching problems," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, pp. 695–703, Sept 1988.

[7] H. Ling and D. jacobs, "Shape classification using the inner-distance," in IEEE Transaction on pattern analysis and machine intelligence, vol. 29, no. 2, pp. 286–299, 2007.

[8] G. E. da Silva and A. R. Backes, "Characterizing 3D shapes: a complex network-based approach," in Proc. of European Signal Processing Conference (EUSIPCO), 2018, pp. 608–612.

[9] G. Mori, S. Belongie, and J. Malik, "Efficient shape matching using shape contexts," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pp. 1832–1837, 2005.

[10] B. Taati and M. Greenspan, "Local shape descriptor selection for object recognition in range data," Computer Vision and Image Understanding, vol. 115, no. 5, pp. 681–694, 2011.

[11] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in Proc. in the International Conference on Intelligent Robots and Systems. IEEE, 2008, pp. 3384–3391.

[12] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in Proc. in the International conference on robotics and automation. IEEE, 2009, pp. 3212–3217.

[13] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," International Journal of Computer Vision, vol. 116, no. 1, pp. 66–89, 2016.

[14] X.-F. Hana, J. S. Jin, J. Xie, M.-J. Wang, and W. Jiang, "A comprehensive review of 3D point cloud descriptors," arXiv preprint arXiv:1802.02297, 2018.

[15] I. Reppa and E. C. Leek, "Surface diagnosticity predicts the high-level representation of regular and irregular object shape in human vision," Attention, Perception, & Psychophysics, pp. 1–20, 2019.

[16] N. Baker and P. J. Kellman, "Abstract shape representation in human visual perception." Journal of Experimental Psychology: General, vol. 147, no. 9, p. 1295, 2018.

[17] S. O. Murray, B. A. Olshausen, and D. L. Woods, "Processing shape, motion and three-dimensional shape-from-motion in the human cortex," Cerebral cortex, vol. 13, no. 5, pp. 508–516, 2003.

[18] B. Alwaely and C. Abhayaratne, "Graph spectral domain shape representation," in Proc. of European Signal Processing Conference (EUSIPCO), 2018, pp. 603–607.

[19] ——, "Adaptive graph formulation for 3D shape representation," in Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, May 2019, pp. 1947–1951.

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in Advances in neural information processing systems, 2017, pp. 5099–5108.

[21] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in Advances in neural information processing systems, 2018, pp. 820–830.

[22] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in Proc. of the IEEE International Conference on Computer Vision, 2019, pp. 6411–6420.

[23] M. Demirci, A. Shokoufandeh, and S. Dickinson, "Skeletal shape abstraction from examples," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 944–952, 2009.

[24] X. Bai, L. Latecki, and W. Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," in IEEE Transactions on pattern analysis and machine intelligence, vol. 29, no. 3, 2007.

[25] C. Zahn and R. Roskies, "Fourier descriptors for plane closed curves," in IEEE Transactions on computers, vol. 100, no. 3, pp. 269–281, 1972.

[26] A. Aghasi and J. Romberg, "Extracting the principal shape components via convex programming," IEEE Transactions on Image Processing, vol. 27, no. 7, pp. 3513–3528, 2018.

[27] H. Jomma and A. Hussein, "Circle views signature: A novel shape representation for shape recognition and retrieval," Canadian Journal of Electrical and Computer Engineering, vol. 39, no. 4, pp. 274–282, 2016.

[28] J. Xie, F. Zhu, G. Dai, L. Shao, and Y. Fang, "Progressive shape-distribution-encoder for learning 3D shape representation," in IEEE Transactions on Image Processing, vol. 26, no. 3, pp. 1231–1242, 2017.

[29] A. Chaouch, M.and Verroust-Blondet, "A new descriptor for 2D depth image indexing and 3D model retrieval," in Proc. of International Conference on Image Processing, vol. 6. IEEE, 2007, pp. VI–373.

[30] C. Aslan, A. Erdem, E. Erdem, and S. Tari, "Disconnected skeleton: Shape at its absolute scale," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 12, pp. 2188–2203, 2008.

[31] B. Wang, W. Shen, W. Liu, and X. Bai, "Shape classification using tree-unions," in Proc. of Pattern Recognition (ICPR). IEEE, 2010, pp. 983–986.

[32] C. Figueiredo, F. Neto, and C. de Paula, "Contour-based feature extraction for image classification and retrieval," in Proc. of International Conference of the Chilean Computer Science Society (SCCC). IEEE, 2016, pp. 1–7.

[33] I. Setitra and S. Larabi, "SIFT descriptor for binary shape discrimination, classification and matching," in Proc. of International Conference on Computer Analysis of Images and Patterns. Springer, 2015, pp. 489–500.

[34] L. Nanni, S. Brahnam, and A. Lumini, "Local phase quantization descriptor for improving shape retrieval/classification," Pattern Recognition Letters, vol. 33, no. 16, pp. 2254–2260, 2012.

[35] D. Pedronette and R. da Silva, "Shape retrieval using contour features and distance optimization," in Proc. of VISAPP (2). Citeseer, 2010, pp. 197–202.

[36] C. Huang, T. Han, and Z. He, "Multi-scale embedded descriptor for shape classification," Journal of Visual Communication and Image Representation, vol. 25, no. 7, pp. 1640–1646, 2014.

[37] Y. Zheng, B. Guo, Y. Yan, and W. He, "O2O method for fast 2D shape retrieval," IEEE Transactions on Image Processing, vol. 28, no. 11, pp. 5366–5378, 2019.

[38] Z. Han, Z. Liu, C. Vong, Y. Liu, S. Bu, J. Han, and C. Chen, "BoSCC: bag of spatial context correlations for spatially enhanced 3D shape representation," in IEEE Transactions on Image Processing, vol. 26, no. 8, pp. 3707–3720, 2017.

[39] R. Horaud and H. Sossa, "Polyhedral object recognition by indexing," Pattern recognition, vol. 28, no. 12, pp. 1855–1870, 1995.

[40] M. Carcassoni and E. Hancock, "Alignment using spectral clusters." in Proc. of British Machine Vision Conference (BMVC), 2002, pp. 1–10.

[41] T. Cour, P. Srinivasan, and J. Shi, "Balanced graph matching," Advances in Neural Information Processing Systems, vol. 19, p. 313, 2007.

[42] O. Duchenne, F. Bach, I. S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2383–2395, 2011.

[43] F. Zhou and F. De La Torre, "Factorized graph matching," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, no. 99, pp. 1–1, 2015.

[44] L. Landrieu and M. Boussaha, "Point cloud oversegmentation with graph-structured deep metric learning," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7440–7449.

[45] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4558–4567.

[46] B. Alwaely and C. Abhayaratne, "Graph spectral domain feature learning with application to in-air hand-drawn number and shape recognition," IEEE Access, vol. 7, pp. 159 661–159 673, 2019.

[47] T. Martinetz and K. Schulten, "A neural-gas network learns topologies," Artificial Neural Networks, pp. 397–402, 1991.

[48] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," journal of computer vision, vol. 59, no. 2, pp. 167–181, 2004.

[49] F. R. K. Chung, Spectral graph theory. Providence, R.I. : American Mathematical Society, 1997.

[50] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," in IEEE Signal Processing Magazine, vol. 30, no. 3, pp. 83–98, May 2013.

[51] M. Akimaliev and M. Demirci, "Improving skeletal shape abstraction using multiple optimal solutions," Pattern Recognition, vol. 48, no. 11, pp. 3504–3515, 2015.

[52] X. Bai, W. Liu, and Z. Tu, "Integrating contour and skeleton for shape classification," in Proc. of international conference on computer vision workshops, ICCV. IEEE, 2009, pp. 360–367.

[53] S. Husain and M. Bober, "REMAP: Multi-layer entropy-guided pooling of dense CNN features for image retrieval," IEEE Transactions on Image Processing, vol. 28, no. 10, pp. 5201–5213, 2019.

[54] M. Reuter, F. E. Wolter, and N. Peinecke, "Laplace–beltrami spectra as 'Shape-DNA' of surfaces and solids," in Computer-Aided Design, vol. 38, no. 4, pp. 342–366, 2006.

[55] A. Chaudhari, R. M. Leahy, B. L. Wise, N. E. Lane, R. D. Badawi, and A. A. Joshi, "Global point signature for shape analysis of carpal bones," Physics in Medicine & Biology, vol. 59, no. 4, p. 961, 2014.

[56] M. Masoumi and A. Hamza, "Spectral shape classification: A deep learning approach," Journal of Visual Communication and Image Representation, 2017.

[57] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," in ACM Transactions on graphics (tog), vol. 21, no. 4, pp. 807–832, 2002.

[58] J. Xie, Y. Fang, F. Zhu, and E. Wong, "Deepshape: Deep learned shape descriptor for 3D shape matching and retrieval," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1275–1283.

[59] Y. Zhou and F. Zeng, "2D compressive sensing and multi-feature fusion for effective 3D shape retrieval," Information Sciences, vol. 409, pp. 101–120, 2017.

[60] P. Papadakis, I. Pratikakis, T. Theoharis, and S. Perantonis, "Panorama: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval," International Journal of Computer Vision, vol. 89, no. 2-3, pp. 177–192, 2010.

[61] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in Proc. of Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920.

[62] Z. Han, M. Shang, Z. Liu, C. Vong, Y. Liu, M. Zwicker, J. Han, and C. Chen, "Seqviews2seqlabels: Learning 3D global features via aggregating sequential views by RNN with attention," IEEE Transactions on Image Processing, vol. 28, no. 2, pp. 658–672, 2018.

[63] A. Bronstein, M. Bronstein, A. Bruckstein, and R. Kimmel, "Analysis of two-dimensional non-rigid shapes," International Journal of Computer Vision, vol. 78, no. 1, pp. 67–88, 2008.

[64] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing their shock graphs," in IEEE Transactions on pattern analysis and machine intelligence, vol. 26, no. 5, pp. 550–571, 2004.

[65] D. Sharvit, J. Chan, H. Tek, and B. Kimia, "Symmetry-based indexing of image databases," in Workshop on Content-Based Access of Image and Video Libraries,. IEEE, 1998, pp. 56–62.

[66] L. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in Proc.of the Computer Vision and Pattern Recognition (CVPR), vol. 1. IEEE, 2000, pp. 424–429.

[67] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," Machine vision and applications, vol. 19, no. 4, pp. 261–275, 2008.

[68] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," in ACM Transactions on graphics (tog), vol. 28, p. 73, 2009.

BASHEER ALWAELY received the B.Sc and the M.Sc degree in Mechatronic engineering from Baghdad University, Baghdad, Iraq, in 2007 and 2012 respectively. He received the Ph.D degree from The University of Sheffield, Sheffield, U.K advised by Dr. Charith Abhayaratne. He is currently a recipient of the European Research Consortium for Informatics and Mathematics (ERCIM) Post-Doctoral Fellowship (2019-2020) to carry out research at the Department of Computer Science at the Norwegian University of Science and Technology (NTNU), Norway. His research interests lie in computer vision and machine learning. http://orcid.org/0000-0002-1845-2641

CHARITH ABHAYARATNE (M'98) received the B.E. degree in electrical and electronic engineering from The University of Adelaide, Australia, in 1998, and the Ph.D. degree in electronic and electrical engineering from the University of Bath, U.K., in 2002. He was a recipient of the European Research Consortium for Informatics and Mathematics (ERCIM) Post-Doctoral Fellowship (2002-2004) to carry out research at the Centre of Mathematics and Computer Science (CWI), The Netherlands, and the National Research Institute for Computer Science and Control (INRIA), Sophia Antipolis, France. He is currently a Lecturer with the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K. His research interests include multidimensional signal processing, image and video compression, visual content understanding, multimedia content security and forensics. He has published more than 80 peer reviewed papers in leading journals, conferences and book editions. Currently, he serves as an associate editor for IEEE Transactions on Image Processing, IEEE Access and Elsevier Journal of Information Security and Applications (JISA).