
Article:
Alwaely, B. and Abhayaratne, C. (2023) GHOSM : Graph-based Hybrid Outline and Skeleton Modelling for shape recognition. ACM Transactions on Multimedia Computing, Communications, and Applications, 19 (2s). 86. pp. 1-23. ISSN: 1551-6857

https://doi.org/10.1145/3554922

GHOSM: Graph-based Hybrid Outline and Skeleton Modelling for Shape Recognition

BASHEER ALWAELY*, Dept. of Electronic and Electrical Engineering, The University of Sheffield, United Kingdom
CHARITH ABHAYARATNE†, Dept. of Electronic and Electrical Engineering, The University of Sheffield, United Kingdom

An efficient and accurate shape detection model plays a major role in many research areas.With the emergence of more complex shapes in real life applications, shape recognition models need to capture the structure with more effective features in order to achieve high accuracy rates for shape recognition. This paper presents a new method for 2D/3D shape recognition based on graph spectral domain handcrafted features, which are formulated by exploiting both an outline and a skeleton shape through the global outline and internal details. A fully connected graph is generated over the shape outline to capture the global outline representation while a hierarchically clustered graph with adaptive connectivity is formed on the skeleton to capture the structural descriptions of the shape. We demonstrate the ability of the Fiedler vector to provide the graph partitioning of the skeleton graph. The performance evaluation demonstrates the efficiency of the proposed method compared to state-of-the-art studies with increments of 4.09%, 2.2% and 14.02% for 2D static hand gestures, 2D shapes and 3D shapes, respectively.

CCS Concepts: • **Computing methodologies** → **Shape representations**; **Hierarchical representations**; **Object recognition**; • **Mathematics of computing** → **Graphs and surfaces**.

Additional Key Words and Phrases: Graph Matching, Spectral Graph Partitioning, Static Hand Gesture.

ACM Reference Format:
Basheer Alwaely and Charith Abhayaratne. 2022. GHOSM: Graph-based Hybrid Outline and Skeleton Modelling for Shape Recognition. *J. ACM* 1, 1, Article 1 (August 2022), 23 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Shape recognition in images, video, point clouds and depth data has often played an important role in computer vision and human–computer interaction (HCI) applications, such as, augmented reality [47], hand gesture recognition [63], object manipulation robotics and object recognition [37]. Such real-world applications are often required to distinguish challenging, high similar shapes in intricate detail [35]. Success of shape recognition is heavily dependent on accurate shape representations that consider shape appearance, structure, any occlusions and articulation. Such models also require to recognise shapes from many image modalities, such as, 2D imaging, 3D point clouds and depth imaging. Shape characterisation is often driven by features identified through modelling shape outline and local protrusions. In addition to modeling outlines and protrusions, the skeletal

Authors' addresses: Basheer Alwaely, Dept. of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, United Kingdom, basheer.alwaely@sheffield.ac.uk; Charith Abhayaratne, Dept. of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, United Kingdom, charith@ieee.org.

0004-5411/2022/8-ART1 $15.00
https://doi.org/XXXXXXX.XXXXXXX

descriptions of shapes are well known to provide unique perceptual information for shape in object recognition task [9].

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 [14, 18, 46] and the importance of local details for the visual perception of shapes [12, 64]. A recent study on human vision suggests that the visual cortex perceives and understands shapes by representing the shape boundary as a connected set of nodes [42]. Inspired by these human vision literature, our recent work demonstrated how to consider a shape as a connected graph to capture both the global outline [3] and the protrusions for shape representation[5].

Graphs have been proven to be an effective means for signals or data captured on an irregular sampling grid. Graphs contain nodes connected by edges that form the relationship between nodes. The structural characteristics of a graph is often modelled by considering the connectivity of the graph nodes leading to so called graph spectral representations[53]. The graph spectral bases depend on the relative measurements between the nodes, making them invariant to the rotation changes. Graphs have the ability to represent shapes with a few nodes and maintain the characteristic of shape. Graph-based representation usually generates graphs based on the shape abstract to simulate the structure followed by either approximate or bipartite matching methods that are used for identification [26]. This has posed many challenges, such as, finding an optimal corresponding match between two graphs [23] and matching two graphs with different numbers of nodes [52, 59].

Recently, handcrafted graph spectral domain features have been used to represent the boundary of the shape [3, 5]. In contrast, the present manuscript considers the skeleton of the shape and proposes new graph spectral domain features of the skeleton formulated as a graph for shape recognition. With this, we propose Graph-based Hybrid Outline and Skeleton Modelling for Shape Recognition (GHOSM). In this approach, we combine graph spectral domain handcrafted features of the shape outline and graph spectral domain handcrafted features of the shape skeleton to capture the global outline and the structural details of the shape, respectively. We show how the graph spectral partitioning of the skeleton graph is exploited to capture structural details in the skeleton-based features. Both outline and skeleton feature sets are concatenated to obtain the overall feature vector followed by a machine learning process to train a classifier. The main contributions of this paper include:

(1) Proposal of a new graph spectral domain representation of shape global outline and structural features using the shape outline and the skeleton.
(2) Proposal of a new skeleton spectral partitioning methodology for obtaining shape structural details.

Although there have been significant developments in recent years in deep learning techniques for shape recognition [34, 40, 44, 56], they often result in high computational complexity models. In this work, we propose handcrafted features as opposed to learned features, as they are more suitable for small datasets with fast training, less computational complexity and competitive recognition accuracy rates. The proposed method was evaluated compared to the state-of-the-art using eleven publicly available datasets: four 2D static hand gesture shape datasets; three other 2D shape datasets; and four 3D shape datasets, as shown in Section 4. The early results of this work, just focussing only on static hand gesture recognition, was presented as a conference paper [4]. In contrast, our argument in the current manuscript is that an appropriate graph partitioning method can enhance the model performance and it can be further extended to be a general 2D/3D shape recognition method. The main differences between the proposed method and our previous works are as follows: Firstly, the present work proposes a new adaptive graph formulation method based on an entropy

computation inspired energy function rather than the conditional connectivity as in [4] for graph partitioning. Secondly, we extend the method to 2D/3D shape recognition in general as well as hand gesture recognition. Thirdly, the performance evaluation is performed on a large number of various datasets. We also provide more insight and details of the proposed method and its performance.

The rest of the paper is organised as follows: the related work of is discussed in Section 2. Section 3 presents the proposed method in detail including graph concepts, the outline and skeleton representations, and the proposed features. The performance evaluation of the proposed method using publicly available shape datasets are presented in Section 4 followed by the conclusions presented in Section 5.

2 RELATED WORK

In this section, we briefly present the existing works for shape recognition. 2D/3D shape recognition works can be broadly categorized into five methods as follows:

2.1 Deep learning-based methods

Recent years have witnessed a trend of using deep learning techniques for shape recognition such as Multi-view CNN [56], VoxNet [40], PointCNN [34], and PointNet [44]. Deep learning methods provide high levels of local details as well as the general appearance through a sequence of convolution layers. The initial layers produce a set of learned features of the shape abstract. With more deep layers and filters, rich knowledge of local details such as edges and protrusions are detected. Deep learning frameworks have demonstrated superior performance for object recognition in general. However, they generally need multi-GPU support for training and relatively large memory space to store the network parameters. This may pose an issue on emerging mobile device platforms, which are limited in computational power and storage.

2.2 Model-based methods

These studies mainly aim to characterize the geometric details of the shape. They can be classified into two types: skeleton and contour representations. In the skeleton-based representation, a tree model is constructed to form a shape descriptor. Then, the similarity is measured based on tree matching. The descriptors are created using short-cut [36], corresponding points [21], skeleton pruning [10] and shape scaling [8]. In the contour-based representation, the outline is formed as a closed curve. This is followed by feature extraction based on the boundaries. Different features are used for the matching such as convex details [1], Fourier descriptors [68] and the chordal axis transform [67]. The main drawback in model-based methods is that the local structural details are omitted or completely ignored in these models.

2.3 View-based methods

View-based studies use different view angles to measure the similarity between shapes. Typical view-based approaches include circle view signature (CVS) [29], multi-view depth line (MDLA) [16], complex function [60] and top-bottom-side views [7]. The main issue in these approaches is in computing the view similarity of different topology samples.

2.4 Feature-based methods

These methods rely on extracting a set of distinguishing features of the shapes. Feature-based methods typically require more than one set of features to describe a complex structure. Such features may include: a Scale Invariant Feature Transform (SIFT) [61], virtual retrieval [69], bag of words [27], variable-dimensional local shape descriptors (VD-LSD) [57], point feature histogram (PFH) [49] and fast point feature histogram (FPFH) [48].

2.5 Graph-based methods

Graph-based models usually generate a graphical model to imitate the shape configuration with low dimension. Approximate and bipartite methods are used to find the optimal map between two set nodes. The approximate method explores the probability of correspondence mapping between two samples through the weight matrix based on: polynomial characterization, center of clusters, spectral relaxation, and the graph incident matrix [15, 19, 28, 71] respectively. Bipartite graph matching is a fast approach where the edges are organized in such a way that no two edges share the same end node. A bipartite graph matching is performed based on: shortest edges and largest eigenvalue [32, 51] respectively. The main issue in graph-based approaches is the implementation time, which makes these methods unsuitable for real-time applications. To address this issue, this paper proposes feature-to-feature assignments, which are based on the geometric structure properties of the objects. The proposed GHOSM combines both feature-based and graph-based styles.

The graph node connectivity is defined as the number of nodes that one node is connected with. The existing methods of connectivity can be categorized into three types:

(1) Special connectivity: for specific applications, nodes have their own connectivity without ability to change it. For example, Minnesota graph [53], where the edges represent the road network.
(2) Full connectivity: each node is connected to all nodes (N) in the graph [3], which means that each node has N-1 connections. Fully connectivity provides a global information about the structure because it preserves the topology of the shape in terms of relative measurement (*e.g.*, Euclidean distance).
(3) K-Nearest Neighbour: where vertices are linked to the nearest K-vertices, and each vertex has K connections [24]. This type of connectivity is usually applied in a uniform grid such as image-based applications.

In contrast to the existing work, in this paper, we formulate an adaptive graph connectivity to fit the geometric details. This adaptive connectivity varies from sample to sample based on their structural characteristics.

3 THE PROPOSED GHOSM METHODOLOGY

GHOSM begins by extracting the outline and skeleton of the shape. Candidate nodes are selected from the contour to form a fully connected graph, which characterizes the topology of the shape. A skeleton representation is employed to partition the shape into meaningful partitions, which helps representing the structural details. At the last step, a combination of outline and skeleton features are used to classify shapes using machine learning techniques. Details of each step are provided in the following subsections. The full pipeline of the proposed method is shown in Fig. 1. In order to provide full details of GHOSM, we start with the preliminaries of the graph signal processing relevant to our work.

3.1 Graph preliminaries

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$ be an undirected graph, where $\mathcal{V}$ is the set of N vertices or nodes, $\mathcal{E}$ is the set of edges and $\mathbf{A}$ is the adjacency matrix with edge weights. We consider $\mathcal{E}$ as the Euclidean distance between nodes and it is invariant to rotation changes. We define the weight, $\mathbf{A}_{i,j}$ corresponding to an edge, $\mathcal{E}_{i,j}$ connecting vertices i and j as follows:

$$\mathbf{A}_{i,j} = \frac{|\mathcal{E}_{(i,j)}|}{\frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|\mathcal{E}_{(i,j)}|}. \tag{1}$$

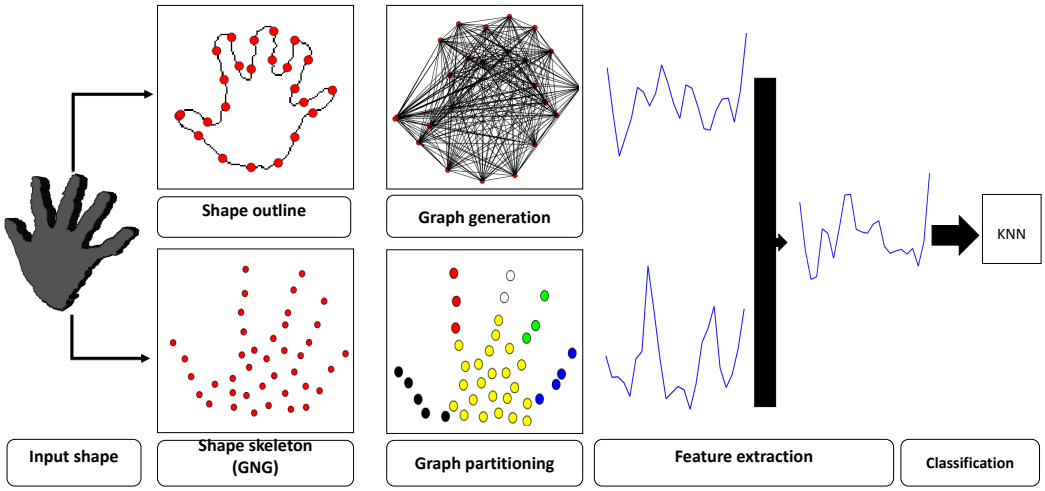


Fig. 1. The pipeline of shape representation includes outline and skeleton representations. The top path shows the outline representation and graph generation to extract the features. The bottom path shows the graph partitioning of the skeleton.

We define the signal $\mathbf{r} : \mathcal{V} \rightarrow \mathbf{R}$, where the i^{th} component represents the Euclidean distance from the center (0,0,0) to the vertex i in $\mathcal{V}$ as follows:

$$\mathbf{r}_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \qquad i = 0, 1, ..., N-1. \tag{2}$$

The combinatorial graph Laplacian matrix is then calculated as follows:

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

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

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

We also define the symmetric normalized Laplacian matrix ($\mathcal{L}$) and the geometric graph Laplacain matrix (Γ) as follows:

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}, \tag{5}$$

$$\Gamma = \mathbf{D}^{-1}\mathbf{A}. \tag{6}$$

Since $\mathbf{L}$ is a symmetric positive semidefinete matrix, 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, leading to an eigenvalue decomposition of $\mathbf{L}$ matrix as follows:

$$\mathbf{L} = \mathbf{U}^t\Lambda\mathbf{U} = \sum_{\ell=0}^{N-1} \lambda_\ell \mathbf{u}_\ell \mathbf{u}_\ell^t, \tag{7}$$

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

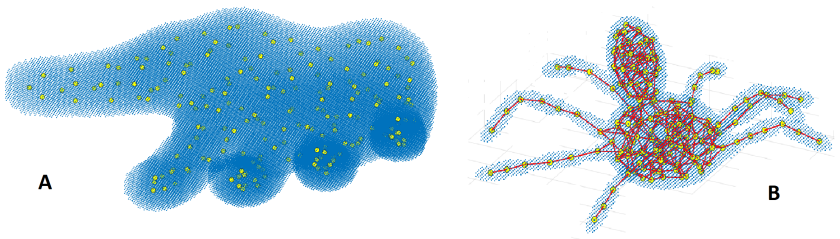


Fig. 2. 3D shape representation using GNG. A: node distribution inside the shape. B: our proposed connectivity.

3.2 Outline and skeleton extraction

2D shapes are usually given as binary images in the data sets. The 2D shape outline is extracted using an edge detector (*e.g.*, Sobel filter). The resulting contour path, P, is usually a smooth curve with N number of pixels, which is different according to the image size and resolution. To reduce the complexity of the subsequent graph spectral decomposition, we choose n_1 number of nodes, where $n_1 < N$, to form a new down-sampled shape contour, $\hat{P}$, as follows (Fig. 2 A):

$$\hat{P}(k) = P\left(\left\{\frac{Nk}{n_1}\right\}\right), \tag{8}$$

where $k = 0, 1, ..., n_1-1$ is the new node index and $\{.\}$ is rounding to the nearest integer operator. $\hat{P}$ is then used as the nodes of the 2D shape. Node ordering is implemented from left to right.

3.3 Outline-based features

3D shapes are often provided as a large point cloud to represent the surface (P). In order to reduce points in the 3D space and to generate a skeleton representation, we use the Growing Neural Gas (GNG) algorithm [38]. GNG provides an excellent quality representation of the shape with a fewer number of nodes. It is an unsupervised procedure to select the optimal nodes to represent the shape based on the distance. Initially, GNG randomly selects two nodes (*i.e.*, $n_2 = 2$), where n_2 is the number of nodes generated by the GNG. Then, based on the probability density, it finds the nearest node to both initial nodes in each iteration (t) as follows:

$$s = argmin \mid\mid w_i(t) - P_t \mid\mid, \qquad i = 0, 1, ..., n_2 - 1, \tag{9}$$

in which w_i represents the weights assigned to each node and s represents the Euclidean distance. The edges between these nodes will be updated based on the error function (e), which represents the difference in distance as follows:

$$e_s(t+1) = e_s(t) + \mid\mid w_s(t) - P_t \mid\mid^2. \tag{10}$$

These steps are repeated until the n_2 nodes are selected. Note that GNG produces unnecessary edges outside the shape surface, for example, linking different fingers outside the geometric representation. Therefore, we consider only coordinates of the nodes, ignoring the connected edges generated by GNG. At the end of the training process, the GNG should satisfactorily cover the shape regions as can be seen in Fig. 2. Since GNG selects nodes regularly based on an unsupervised optimization process in a way that these nodes have a uniform distribution inside the shape, the noisy pixels are removed by the GNG process. The node ordering is implemented from bottom to top, and then left to right.

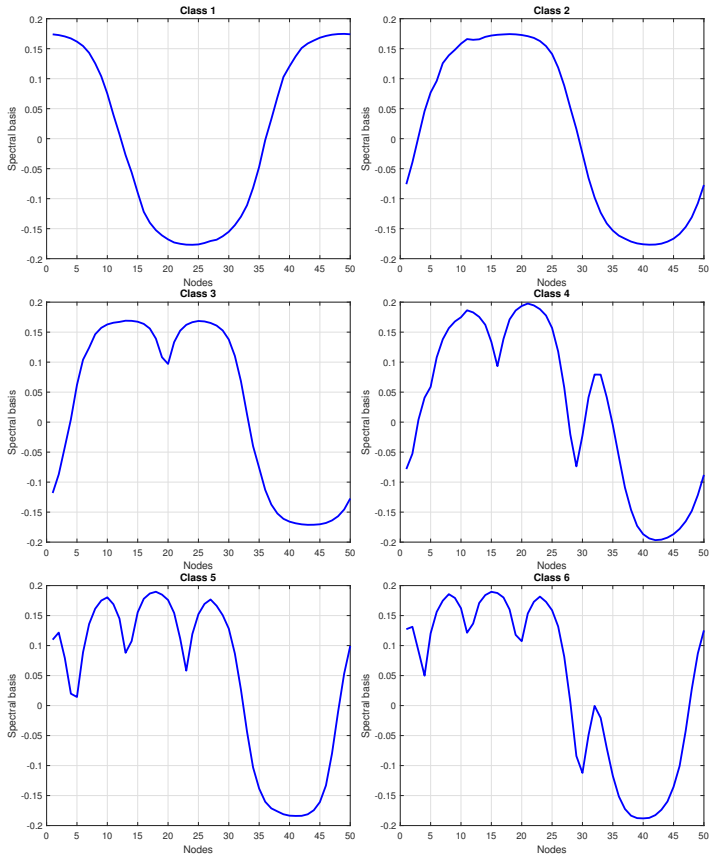
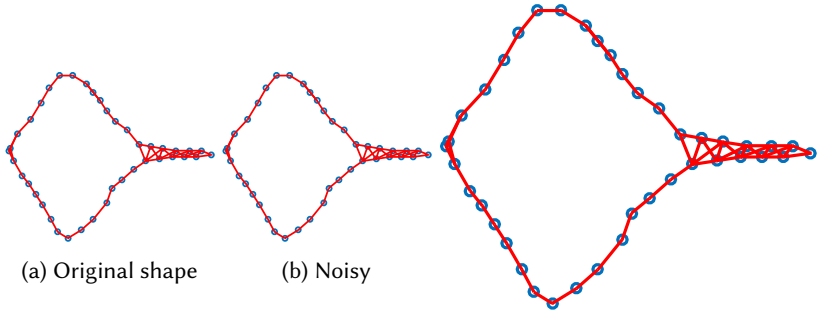


Fig. 3. Graph spectral response (*i.e.*, the second eigenvector) to different classes using dataset (**d1**). From left to right, top row (classes 1-3) and the bottom row (classes 4-6) .

A fully connected graph is generated over the outline, and its spectral features are extracted for classification. Our proposed method specifically exploits the graph eigenvectors and eigenvalues. To show the importance of their characteristics, four experiments are conducted to evaluate different parameters as follows:

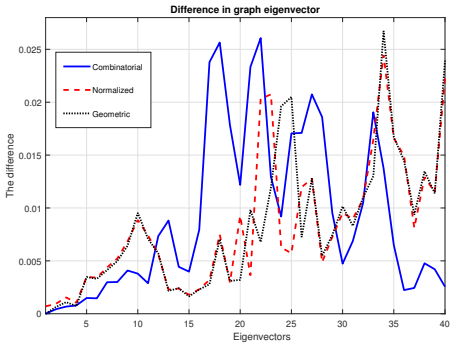
3.3.1 The spectral response: The probability of algebraic connectivity is inversely proportional to the weight A_{ij} of the edge connecting i and j in the adjacency matrix [53]. According to this concept, we expect to see high values in the spectral bases of a set of nodes that are close together compared to other regions. This can be clearly seen in the extended fingers in hand gesture recognition and limbs or other small parts in shapes. Fig. 3 shows the second eigenvector of $\mathcal{L}$ for classes from 1-6 using the dataset **d1** as an example (more details about the datasets are presnted in Section 4.1). We can see that the peaks in the spectral bases refer to the extended fingers. The sequence of the peaks and troughs helps to localize the extended fingers with respect to the full contour details. This can be explored in classifying shape that have the same number of the extended parts.

3.3.2 The effect of noise: Graphs $\mathcal{G}$ and its noisy version $\mathcal{G}'$ have similar eigenvectors $\mathbf{U}$ and $\mathbf{U}'$, respectively. Experimentally, the similarity between the graph (Fig. 4a) and its noisy version (Fig. 4b) can be shown to be higher in high-frequency eigenvectors (Fig. 4d). This is true because the

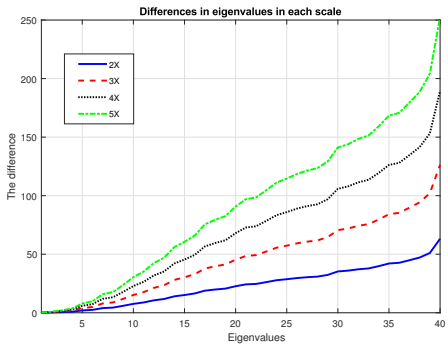


(a) Original shape (b) Noisy

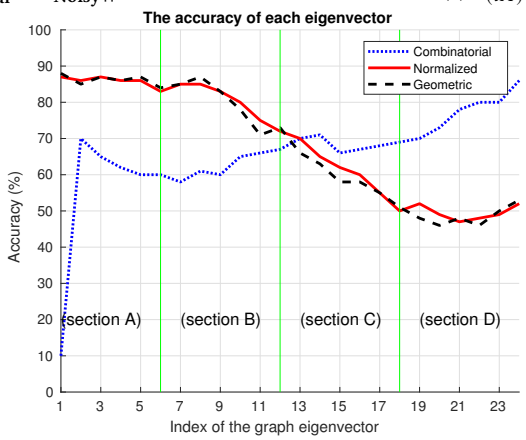
(c) Double size



(d) $||\mathbf{U}_{\mathbf{Original}} - \mathbf{U}_{\mathbf{Noisy}}||^2$



(e) $\lambda_{(iX)} - \lambda_{(1X)}$



(f) The accuracy score for individual graph eigenvectors using **d1**.

Fig. 4. Understanding the effect of noise and size changes on graph spectral bases.

low-frequency bases correspond to the global outline representation and the high-frequency bases correspond to the representation of more structural details with local variations [53]. In other words, low-frequency bases are less affected by noise.

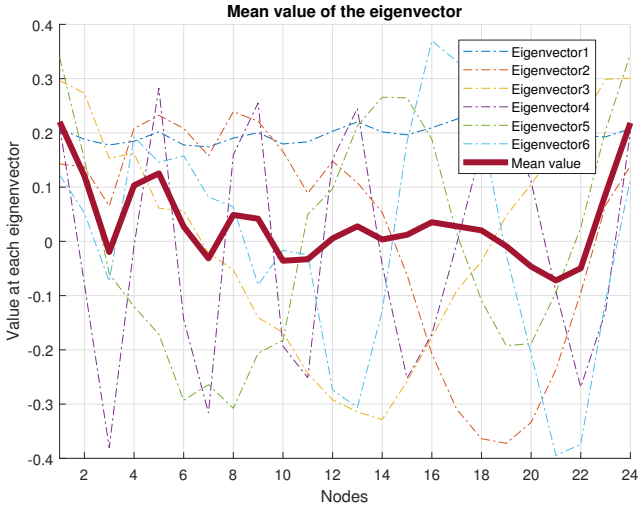


Fig. 5. Mean value of the basis.

3.3.3 Scale changes: Since the graph eigenvectors are presented as scalar values in the range of $[-1, 1]$ to reflect the relative measurements between nodes, they are not affected by linear changes. Scaling changes can only be captured by the eigenvalues. For illustration, an experiment has been conducted to test the graph eigenvalues (Fig. 4a) and those of a scaled graph (Fig. 4c). As expected, these two graphs have exactly the same eigenvectors but different eigenvalues. In other words, $\mathbf{u}_{(0)} = \mathbf{u}'_{(0)}, \mathbf{u}_{(1)} = \mathbf{u}'_{(1)}$, and so on. Fig. 4e shows the differences between the eigenvalues of $\mathcal{G}$ and many of its scaled versions. The big difference always occurs in the last eigenvalues. Therefore, the last λ_{n_1-1} is used in GHOSM to capture the shape size.

3.3.4 The most effective bases: In the final experiment, we test the eigenvectors to determine the most effective bases for shape representation. Each eigenvector is used as a feature to be classified. As can be seen in Fig. 4f, a wide range of accuracy levels is achieved and it can be categorized into four sections (A, B, C, D). Both normalized and geometric bases recognize shapes with a high level of accuracy in the section A. Then, it drops towards the section D. The combinatorial version provides inverse behavior and it reaches the highest accuracy in the section D. This experiment shows that the A and D sections provide a significant contribution of the matching percentage, whereas, the rest of the eigenvectors (*i.e.*, sections B and C) show lower range of accuracy. Therefore, this paper employs a certain range of the bases to eliminate inefficient eigenvectors for shape recognition. To address the ambiguity of the sign, each eigenvector ($\mathbf{u}_\ell$) is updated as follows:

$$\mathbf{u}_{(\ell)} = \begin{cases} \mathbf{u}_{(\ell)} & if \quad \mathbf{u}_{(\ell,\mathbf{0})} \geq 0, \\ \alpha \mathbf{u}_{(\ell)} & if \quad \mathbf{u}_{(\ell,\mathbf{0})} < 0, \end{cases} \tag{11}$$

where $\alpha = -1$. Let $\mathbf{S}_{(\mathbf{n_1},\eta)}$ is an array containing the eigenvectors in x-section, where η is the number of eigenvectors in that section. Then, we compute $|S|$ as follows:

$$|S_i| = \frac{1}{\eta} \sum_{j=0}^{\eta-1} \mathbf{u}_{(\mathbf{i},\mathbf{j})}, \qquad i = 0, 1, ..., n_1 - 1. \tag{12}$$

$|S|$ is the mean of the eigenvectors in the selected section with length n_1. Fig. 5 shows an example of these bases and their mean value. Then, the global features (f_G) of GHOSM based on the outline

representation are obtained as,

$$f_G(x) = \mathbf{r}\ e^{(\lambda_{(1)} - \lambda_{(n_1-1)})}|S_x|. \tag{13}$$

In order to improve the feature representation for the three Laplacian matrices, we consider the accuracy status in Fig. 4f. Therefore, the outline-based global features are computed as shown in Eq. (14), Eq. (15), and Eq. (16) for the combinatorial, normalized, and the geometric Laplacian versions, respectively.

$$f_G(L) = \mathbf{r}\ e^{(\lambda_{(1)} - \lambda_{(n_1-1)}/1000)}|S_D|, \tag{14}$$

$$f_G(\mathcal{L}) = \mathbf{r}\ e^{(\lambda_{(1)} - \lambda_{(n_1-1)})}|S_A|, \tag{15}$$

$$f_G(\Gamma) = \mathbf{r}\ e^{(\lambda_{(1)} - \lambda_{(n_1-1)})}|S_A|. \tag{16}$$

The length of the outline features is n_1. These features combine spatial (*i.e.*, $\mathbf{r}$) and spectral details (*i.e.*, S_x and λ) of the shape. They also provide an efficient representation of shapes. However, a mismatching problem occurs when the outline of different samples is conceptually similar. Therefore, we need to consider more details of the structure. One way to do this is by simplifying the structure of the shape skeleton into several partitions as shown in Section 3.4.

3.4 Skeleton-based features

In this subsection, we provide a fast partitioning method to simplify the structures for efficient and reliable matching. Graph partition has a long history in computer vision, specifically using the eigenvector corresponding to the smallest non-zero eigenvalue, which is known as the Fiedler vector. It provides the minimum cutting ratio according to the optimization formula in Eq. (17) [25],

$$\lambda_1 = min\left(\frac{\mathbf{U}^t\mathbf{L}\mathbf{U}}{\mathbf{U}^t\mathbf{U}}\right). \tag{17}$$

The vast majority of current partitioning methods require determining the number of clusters in advance. To solve this issue, we propose new rules to achieve a fully automatic and stable recursive hierarchical partitioning, leading to automatically identifying the meaningful parts of the structure without any human intervention.

For graph partitioning, the main differences in terms of graph generation is that we use the combinatorial Laplacian version in Eq. (3) and consider the adaptive connectivity proposed in this section. The main reason for using the combinatorial Laplacian matrix is its ability to efficiently detect the local details. Further, our propose partition method is based on the adaptive connectivity to achieve a stable segmentation. While the smallest distance ($t_\circ$) was used to connect all nodes as a single set in [5], the present work increases the stability of the division process by considering a certain distance to link nodes. This distance depends on the topology of the shape and varies from sample to sample. The procedure can be summarized as follows:

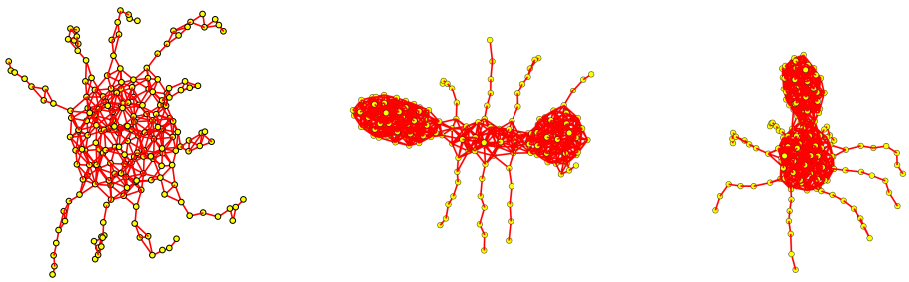
(1) For a given shape, we initially generate a graph based on the initial edge threshold distance, $t_\circ$.
(2) The threshold distance is increased with a chosen step size (δ) for n_2 times.
(3) At each iteration, the energy function ($\boldsymbol{E}_\delta$) of the normalized node degree ($\mathcal{B}_i^\delta$) at node, i, for distance, $t_\circ + \delta$, is computed as follows:

$$\boldsymbol{E}_\delta = \sum_{i=0}^{n_2-1} \mathcal{B}_i^\delta \log_2\left(\frac{1}{\mathcal{B}_i^\delta}\right). \tag{19}$$

(4) We use the distance ($T = \text{argmax}(\boldsymbol{E})$) that provides the maximum energy to generate the graph with adaptive connections and extract its features.

Algorithm 1 Computing the adaptive connectivity value

Inputs: Unconnected 3D point cloud graph $\mathcal{G}$.
Outputs: The distance required to connect the graph nodes.
$k \leftarrow$ Number of graphs in the dataset.
$n_2 \leftarrow$ The number of the nodes generated by GNG.
$E \leftarrow$ Energy function.
for $i = 0 : (k-1)$ **do**
    $\hat{P}_i \leftarrow$ Graph representation as (x_i,y_i,z_i).
    $t_\circ \leftarrow$ Smallest distance to connect all nodes as one set.
    for $\delta = 0 : (n_2 - 1)$ **do**
        $\Phi^{\delta} \leftarrow$ Node degree of $\hat{P}_i$ using $(\delta + t_\circ)$.
        $\mathcal{B} \leftarrow$ Normalizing the node degree $(\frac{\Phi^{\delta}}{\max(\Phi^{\delta})})$.
        $E_\delta \leftarrow \sum_{i=0}^{n_2-1} \mathcal{B}_i^{\delta} \log_2 \left(\frac{1}{\mathcal{B}_i^{\delta}}\right)$.
    end
    $T_i \leftarrow \text{argmax}(E)$.
end



(a) Crabs (b) Ants (c) Spider

Fig. 6. Adaptive connectivity of different samples.

Algorithm 1 briefly summarizes the procedure. The energy function (E) is inspired by the computation of entropy in information theory. We do not call it entropy, since it does not involve probabilities. The entropy is usually regarded as a measure of randomness. The same concept is used here to capture the description of the shape by considering the node connectivity. This benefits in strengthening and weakening the node connections according to the shape structure. The higher the variations in the node vector the higher the E_δ value for a given δ and vice versa. Fig. 6 shows examples of the adaptive connectivity. For example, the connectivity of the limbs in the three shapes always has less connected nodes compared to the other nodes in the shape.

After the graph is generated based on the adaptive connectivity, the Fiedler vector is used for partitioning the graph into two sub-graphs based on its sign. In other words, we spread the nodes into two groups based on the sign of the Fiedler vector. For example, it can result in a sub-graph (g1) consisting of nodes with a positive sign and a sub-graph (g2) consisting of nodes with a negative sign. For each sub-graph, we compute the Fiedler vector again and repeat the procedure for partitioning into the new two sub-graphs. This process is repeated until the complete segmentation is achieved. In the literature, Bayesian probability has been widely used for partitioning and graph cut [33]. However, these methods suffer from conciseness and accuracy as probability parameters depend entirely on the training samples. In contrast, the spectral graph domain approach proposed in this work provides a more stable representation. In order to avoid fragmentation in the main body and in the small parts in a shape, the segmentation process should satisfy the following two rules:

(1) The minimum number of nodes in each sub-group should be 2.

$$n_{g_1} \geq 2, \tag{20}$$

$$n_{g_2} \geq 2, \tag{21}$$

where n_{g_1} and n_{g_2} are the number of nodes in the new generated sub-graphs.

(2) In order to generate two sub-graphs (g_1, g_2), the difference in number of nodes between them must be $> n_2/3$, where n_2 is the total number of nodes in the skeleton representation. *i.e.*,

$$|n_{g_1} - n_{g_2}| > n_2/3, \tag{22}$$

Fig. 7 illustrates nine levels of 3D segmentation with the corresponding number of nodes.

Finally, we propose a skeleton-based structural feature vector, f_L, comprising of the following four components: f_a, f_b, f_c and f_d.

(1) The first part of the feature vector addresses the local details by using the normalized node degree $\mathcal{B}$ at the adaptive connectivity:

$$f_a = \mathcal{B}. \tag{24}$$

(2) In the second part of the feature vector, we include the features from the 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) as shown in Eq. (2). Although r_i represents the global shape, in order to improve the discrimination among classes by considering the local variations, we modulate r with corresponding eigenvalues λ_i corresponding to the graph formulated with adaptive connections for the given shape sample, as follows:

$$f_b = \lambda_i \, \mathbf{r}_i \qquad i = 0, 1, ..., n_2 - 1. \tag{26}$$

(3) The third part of the feature represents the number of clusters generated using the proposed graph partitioning algorithm:

$$f_c = C, \tag{28}$$

where C is the number of clusters after graph partitioning. For example, C is 9 for the shape in Fig. 7.

(4) The fourth part of the feature shows the number of nodes connected to only one node. This represents the specific shape characteristics, for example, the limbs in the skeleton as shown in Fig. 6:

$$f_d = \psi, \tag{30}$$

where ψ is the number of nodes that have only a single link (*i.e.*, node degree is 1). For example, $\psi = 8$ for the shape shown in Fig. 7.

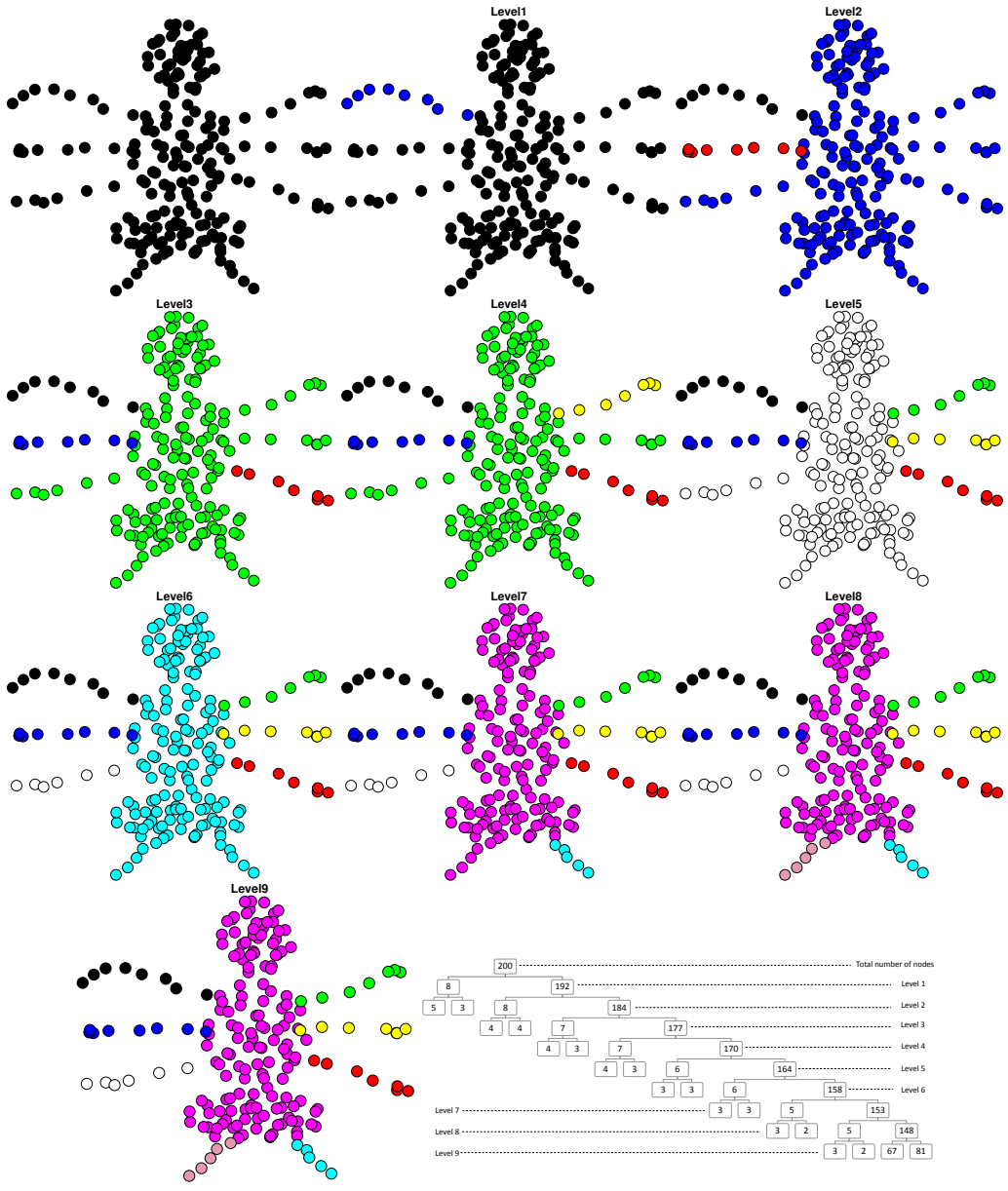


Fig. 7. Graph partitioning with its corresponding number of nodes at each level. In this example, the number of clusters C is 9 in the final level, each cluster is coloured in a different colour.

Concatenating these features (f_a, f_b, f_c, f_d) results in high discriminate local-based features (F_L). The skeleton-based feature length is $2n_2 + 2$. The total length of GHOSM features $(f_G + f_L)$ is $n_1 + 2n_2 + 2$.

3.5 Machine learning

In this final step, machine learning is used to train a classifier based on the feature vectors generated in Section 3.3 and Section 3.4. We have evaluated several classifiers including Support Vector Machine with a cubic form as a kernel function (CSVM), Classification Tree (CT), Discriminant Analysis (DA), the Nearest Neighbour (KNN), 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 4.

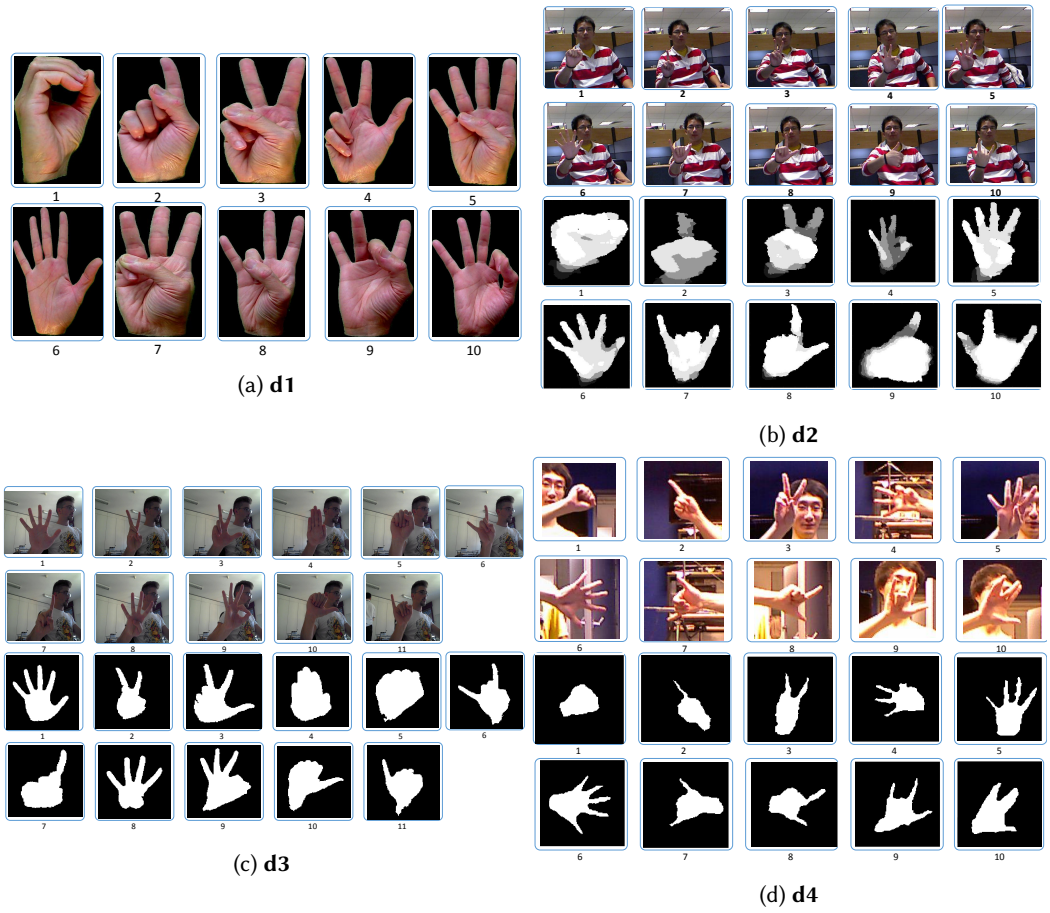
4 PERFORMANCE EVALUATION

This section presents information on the experimental set up of the proposed method including, the datasets used for evaluation, the experimental set up, the performance of the proposed GHOSM in terms of accuracy rates, confusion matrices, comparison with the existing methods and the ablation studies.

4.1 Datasets

The experiments were based on both 2D and 3D shape datasets. The static hand gesture datasets were used as a special case of 2D shapes. Our experiments included four static 2D hand gesture datasets, three other 2D shape datasets and four 3D shape datasets:

(1) **d1**: ASL dataset [13] consists of 36 American Sign Language (ASL) gestures performed by five persons. Images are captured using a neutral-coloured. We focus on the 10 classes corresponding to the numbering gestures from (0 - 9) with 65 samples for each class as shown in Fig. 8a.
(2) **d2**: NTU static hand gestures recognition dataset [45] consists of 10 subjects × 10 hand gestures × 10 different orientations = 1000 colour and its corresponding depth images as can be seen in Fig. 8b. The database includes the subject poses with various hand orientation, scale and articulation. Only depth images are used in these experiments.
(3) **d3**: This dataset [41] contains 120 samples for 11 classes, which are implemented by 4 persons. The dataset provides RGB images and its corresponding depth image. A confidence depth map for each sample is used for the evaluation as shown in Fig. 8c.
(4) **d4**: HKU dataset [62] contains 100 samples for 10 classes, which are implemented by 5 persons. The dataset provides RGB images and its corresponding depth image. Only the depth information of each gesture is used in the experiments as shown in Fig. 8d.
(5) **d5**: ETU10 silhouette dataset provides a 5 degree rotation difference for each class. Sample silhouettes from each class are shown in the top two rows of Fig. 9a (top). ETU10 silhouette has 10 classes ×72 shapes per class = 720 total images. The ten classes in the confusion matrix correspond to the Bed, Bird, Fish, Guitar, Hammer, Horse, Sink, Teddy, Television and Toilet respectively.
(6) **d6**: Kimia 99 dataset [50] consists of 9 classes × 11 samples = 99 images as shown in Fig. 9a (bottom). The nine classes in the confusion matrix correspond to the Fish, Hand, Human, Aeroplane, Ray, Rabbit, Misk, Spanner and Dog respectively.
(7) **d7**: MPEG-7 CE-Shape-1 PartB (MP7-shape) dataset [30] consists of 70 classes with 20 samples per class, resulting in a total of 1400 2D shapes.
(8) **d8**: SHREC2010 dataset [54] consists of 20 objects × 10 classes = 200 points cloud models in total. These samples are taken from McGill Articulated Shape Benchmark dataset and some of its samples are shown in Fig. 9b. The classes include Ants, Crabs, Hands, Humans, Octopus, Pliers, Snakes, Spectacles, Spiders, and Teddy respectively.



(a) **d1** (b) **d2** (c) **d3** (d) **d4**

Fig. 8. Samples of static hand gestures datasets are used to evaluate GHOSM.

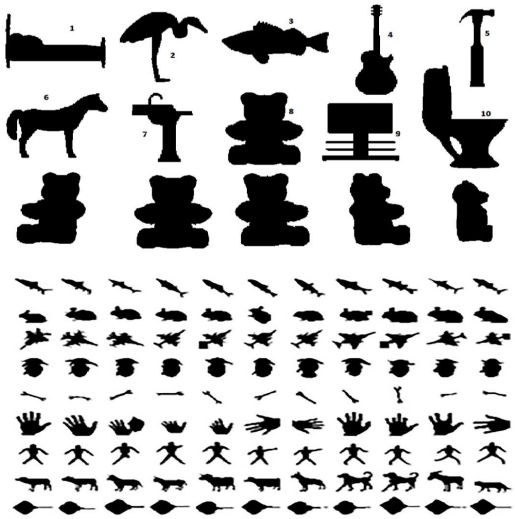
(9) **d9**: 3D shape benchmark dataset [17] provides 19 classes× 20 samples per class = 380 shapes in total. The shapes in each class were presented in different orientations, scales and articulation. These classes include: Human, Cup, Glasses, Airplane, Ant, Chair, Octopus, Table, Teddy bear, Hand, Plier, Fish, Bird, Mech, Bust, Armadillo, Bearing, Vase, and Four Leg respectively. Some of the samples are shown in Fig. 9c.

(10) **d10**: ModelNet10 dataset [65] consists of 10 classes of 3D shapes formed as a CAD model of the point could as can be seen at Fig. 9d. The number of training and validation samples per class ranging 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.

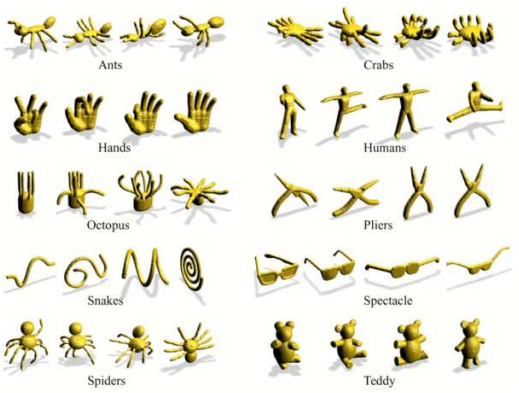
(11) **d11**: ModelNet40 dataset [65] consists of 40 classes with the number of training and validation samples per class ranging from 64-889 and 20-100, respectively.

4.2 Performance evaluation of the proposed GHOSM

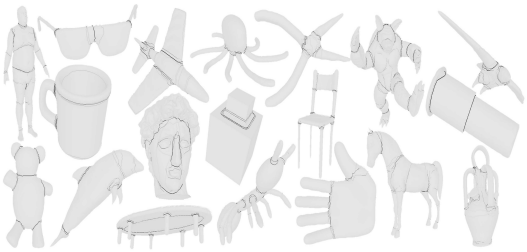
All the experiments were implemented using MATLAB R2019a on a PC with Intel processor CPU@3.6GHz and RAM 16GB. The number of nodes used for 2D shapes representation was $(n_1 = 80, n_2 = 50)$ and 3D shapes representation was $(n_1 = 320, n_2 = 200)$, which were determined



(a) **d5**(top) and **d6**(bottom)



(b) **d8**



(c) **d9**

(d) **d10**

Fig. 9. Samples of 2D/3D datasets are used to evaluate GHOSM.

by experiments. The accuracy rates presented in this paper for most of the datasets are the average accuracy rates obtained from a k-fold validation scheme. The most efficient k value for each dataset was chosen and shown with the results. For most datasets, k=10 was considered as determined by experiments. Since ModelNet datasets (**d10** and **d11**) provide separate training and testing samples, k-fold cross validation was not used for them.

TABLE 1 shows the accuracy rates for the proposed method evaluating the performance of various classifiers. As can be seen, NN and KNN (with the normal Euclidean distance) classifiers result in the best accuracy rates among all classifiers. From TABLE 1, we can conclude that KNN and NN result in the highest accuracy rates and therefore we use these two classifiers for the rest of the evaluations. We recommend KNN as the preferred classifier because it is faster compared to the NN classifier. In addition, for both 2D and 3D shapes, we can clearly see that the proposed GHOST + CT has the worst accuracy rates compared to other methods. The main problem with the classification tree is the lack of a principle probabilistic framework, which leads to poor results. Also, classification trees are extremely sensitive and easily result in over-fitting compared to the other methods.

Table 1. Overall accuracy rates (%) for the GHOSM using different classifiers. Red indicates the highest score.

Dataset		Cross validation (k)	GHOSM + CSVM	GHOSM + CT	GHOSM + DA	GHOSM + KNN	GHOSM + NN
2D (Hand gesture)	**d1**	10	99.1	91.3	98.8	99.6	99.4
	d2	10	98.27	88.11	97.57	99.7	99.7
	d3	10	91.44	71.81	90.15	94	94.4
	d4	10	96.2+	78.1	95.1	99.5	99.4
2D (Other shapes)	**d5**	10	98.8	85.2	96.43	99.58	99.7
	d6	9	97.98	85.86	97.98	100	100
	d7	10	87.8	84.7	85.3	94.2	97.4
3D	**d8**	10	89.5	71.4	92.47	94	94
	d9	10	77.3	64.32	74.12	76.32	75.52
	d10	N/A	79.5	72.8	76.82	84.66	87.11
	d11	N/A	77.1	69.8	63.7	82.74	86.62

Table 2. Overall accuracy rates (%) for the GHOSM compared to the existing work. Red indicates the highest score, and blue is the second highest.

Dataset		Cross validation (k)	Proposed GHOSM + KNN	Proposed GHOSM + NN	Our previous work [4]	Existing handcrafted features based methods	Existing deep learning based methods
2D (Hand gesture)	**d1**	10	99.6	99.4	98.5	98.51[6]	98.40 [58]
	d2	10	99.7	99.7	99.7	99.6 [62]	N/A
	d3	10	94	94.4	93.7	89.91 [41]	94 [22]
	d4	10	99.5	99.4	99.4	99.1 [62]	N/A
2D (Other shapes)	**d5**	10	99.58	99.7	99.1	97.5 [2]	N/A
	d6	9	100	100	97	100 [43]	N/A
	d7	10	94.2	97.4	91.28	96.6[11]	N/A
3D	**d8**	10	94	94	91.5	92.5[31]	96 [39]
	d9	10	76.32	75.52	74.32	70.79 [20]	78.2 [66]
	d10	N/A	84.66	87.11	80.76	73.09 [72]	88.4 [55]
	d11	N/A	82.74	86.62	79.8	N/A	88.93 [70]

4.3 Comparison with the existing methods

TABLE 2 compares the performance of the proposed method with the best published results in the literature. Our proposed handcrafted features (GHOSM) outperform the existing handcrafted features based methods for 10 of the datasets. The differences in performance between GHOSM and the existing handacrafted features based work for **d1** to **d10** are +1.09, +0.1, +4.49, +0.4, +2.2, 0, 0.8, +1.5, +5.53, +14.02m, respectively. Results also show improved performance compared to our previous work in [4]. The confusion matrices with recognition accuracy rates for each class in datasets that provide less than 100% overall accuracy are presented in Fig. 10.

In addition, GHOSM shows excellent performance for 2D static hand gesture datasets, outperforming both handcrafted feature-based and deep learning-based methods. Static hand gesture shapes can sometimes result in the same shape contour as shown in Fig. 11. For example, the hand gestures, fist and open hand have almost the same shape contour which can result in a highly

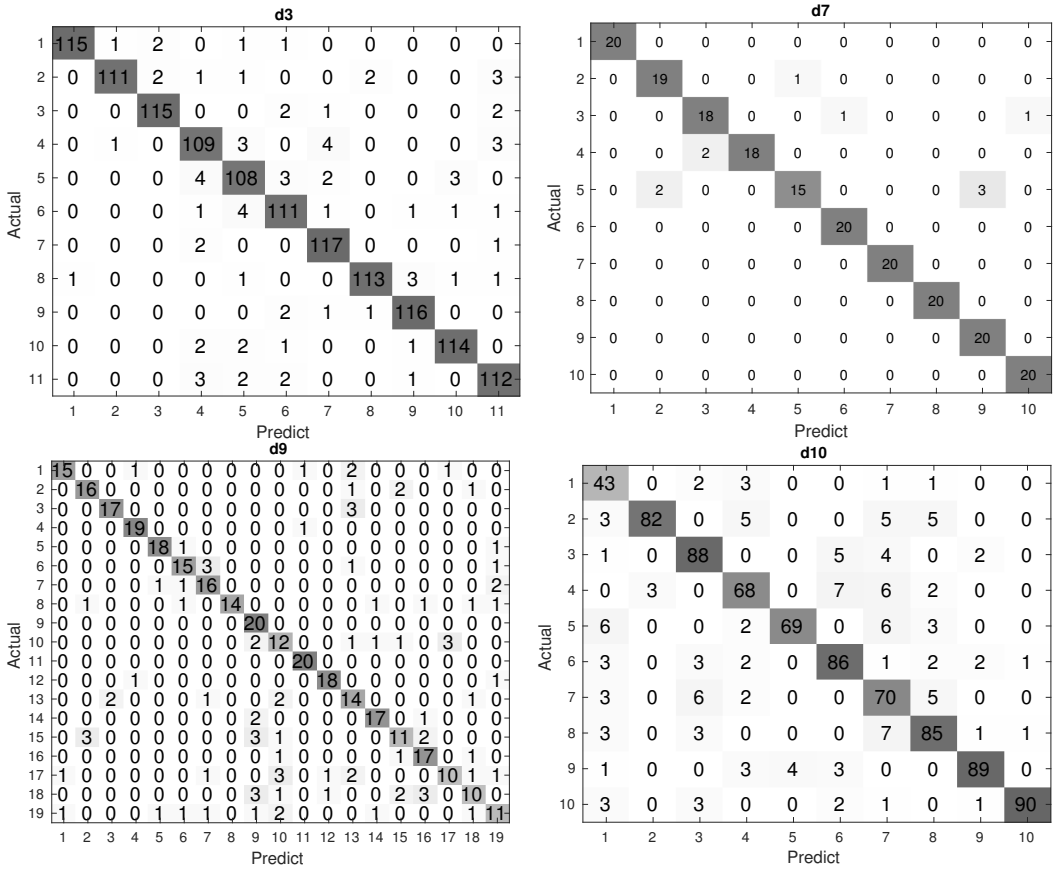


Fig. 10. Confusion matrices of **d3**, **d8**, **d9**, and **d10** based on KNN.

similar segmentation process and a skeleton. The same issue can be observed in class 8 and class 9 in **d1** (as shown in Fig. 8). In these cases, the outline-based global features have worked better than skeleton-based structural features. GHOSM also outperforms deep learning methods reported for **d1** and **d3** datasets.

For 2D datasets, **d5** and **d6** datasets contain highly detailed shapes and GHOSM recognises these classes with high accuracy rates. Despite having shape samples with different angles of views in **d5**, GHOSM outperforms the existing methods. **d7** is one of the most challenging 2D datasets due to having a small number of samples in each class compared to the total number of classes, yet GHOSM with NN outperforms the existing handcarfted features based methods.

For 3D datasets, GHOSM exceeds the performance of existing handcrafted features based methods by 1.5% for **d8**, 5.53% for **d9** and 14.02% for **d10**. GHOSM recognizes all shape classes in **d8** with a high accuracy rate of 94% (Fig. 10). The main confusing class is the octopus class (class 5), which matches spider class (class 9) due to its similarity in graph structure. For example, 3D shapes like Crabs, Ants, and Spider in SHREC2010 have high similarities in their surfaces as can be seen in Fig. 9 (**d8**). **d9** is a very challenging dataset due to various angles of views, and the GHOSM outperforms the existing hand-crafted features based methods by 5.53%. **d10** is a large dataset and usually used to evaluate deep learning methods. GHOSM has achieved an overall recognition accuracy rate

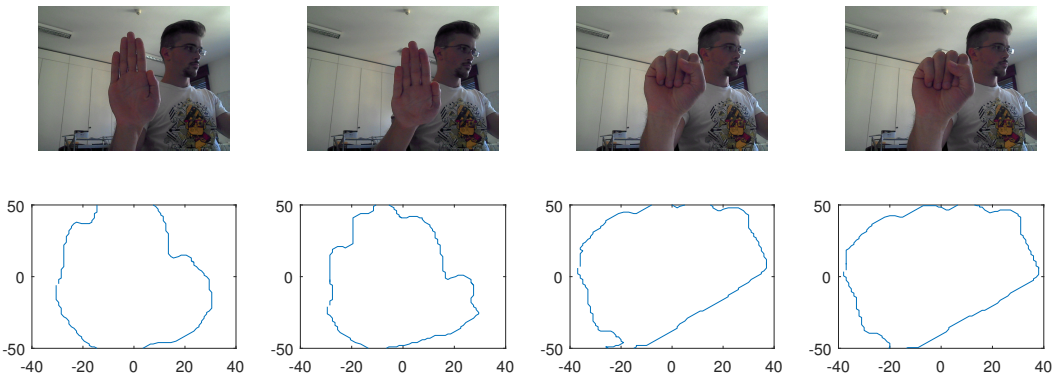


Fig. 11. Confusion cases.

of 87.11% significantly outperforming the other hand-crafted features based methods. The most confusing class is the Night-stand class (class 7) due to its structural similarity to Chair class (class 3) and Desk class (class 4). Errors also appear in the Bathtub class (class 1), which is confused with the Dresser class (class 5).

Although our proposal is on handcrafted features, we have included deep learning methods for comparison. Out of all 2D shape datasets, used in our evaluation, deep learning results has been reported only for **d1** and **d3** datasets in the literature. Our proposed GHOSM, which is based on handcarfted features, has outperformed the deep learning methods by 1%, and 0.4%, respectively. However, for 3D datasets, deep learning based methods have shown 1.29%-2% advantage over the proposed method. As evident from TABLE 2, the existing handcrafted features-based methods have not performed well for 3D datasets. Whereas, GHOSM shows comparable results with those of deep learning based methods that were trained on large datasets with a high computational resources. This is very encouraging for applications that have small datasets and low computational resources.

4.4 Computational complexity

The computational complexity of GNG, partitioning, fully connected graph generation, feature extraction and classification stages are $O(n^2)$, $O(n^2)$, $O(n^2)$ and $O(n^2)$, respectively. Then, the overall complexity of the proposed GHOSM 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 2.4 seconds, which reflects the real-time performance of the proposed method. Although the implementation was in Matlab, the proposed method can be implemented on any platform, programming language and edge devices leading to many real-world applications, such as, automated analysis and understanding of complex shapes, hand gesture analysis, sign language recognition, point cloud analysis, and other human-computer interaction and computer vision applications.

4.5 Ablation studies

We evaluate the performance of GHOSM using the global outline-based features and skeleton-based structural features. TABLE 4 shows that the outline feature representation f_G efficiently recognises different classes. In other words, a fully connected graph based shape model works well to classify hand gestures and 2D shapes. However, 3D shapes often need further structural details for accurate

Table 3. The average time to perform different steps of the proposed method.

Step	Performance average time (ms)
GNG	1984.54
Partitioning	428.12
Fully connected graph generation	0.429
Feature extraction	0.185
Classification	1.654
Full time system	2415 ms ≈ 2.4 seconds

Table 4. The accuracy rate (%) of the outline, skeleton and combination of both.

Dataset	Outline-based Global features	Skeleton-based Local features	Combined (GHOSM)
d1	99.6	91.8	99.6
d2	99.2	89.2	99.7
d3	93	75.3	94
d4	97.8	86.5	99.5
d5	99.1	99.58	99.58
d6	95.8	100	100
d7	86.6	94.2	94.2
d8	85	93	94
d9	61.57	76.32	76.32
d10	78.4	84.25	84.66
d11	70.2	82.74	82.74

recognition. For this reason the 3D shapes have benefited from the inclusion of the skeleton-based structural feature representation in GHOSM.

5 CONCLUSIONS

This paper has proposed a new set of hand-crafted features, known as GHOSM, for shape recognition by modelling both outline-based global features and skeleton-based structural features that are based on spectral partitioning of the underlying graph with shape characteristics driven adaptive connectivity. To achieve this, we have proposed a new method for formulating a graph with adaptive connections to represent shapes' global structure with an unique graph and spectral partitioning of the graph. This is followed by proposing graph spectral features to capture both global outline and structural characteristics of the shape. The effectiveness of the proposed GHOSM was verified by experiments on four static hand gestures datasets, three 2D shape datasets and four 3D shape datasets. The proposed GHOSM, which is a handcarfted features based method, has outperformed the existing handcrafted features-based methods, by increments of up to 4.09%, 2.2% and 14.02% for 2D static hand gesture, 2D shapes, and 3D shapes datasets, respectively. It has also outperformed the deep learning based methods reported in the literature for 2D hand gesture datasets. The accuracy rates for 3D datasets have shown performance comparable to those of deep learning-based approaches.

ACKNOWLEDGMENTS

The work of Dr. Alwaely was supported by the European Research Consortium for Informatics and Mathematics Alain Bensoussan Fellowship Program.

REFERENCES

[1] A. Aghasi and J. Romberg. 2018. Extracting the Principal Shape Components via Convex Programming. *IEEE Transactions on Image Processing* 27, 7 (2018), 3513–3528.
[2] M. Akimaliev and M. Demirci. 2015. Improving skeletal shape abstraction using multiple optimal solutions. *Pattern Recognition* 48, 11 (2015), 3504–3515.
[3] B. Alwaely and C. Abhayaratne. 2019. Graph Spectral Domain Feature Learning With Application to in-Air Hand-Drawn Number and Shape Recognition. *IEEE Access* 7 (2019), 159661–159673.
[4] B. Alwaely and C. Abhayaratne. 2019. Graph Spectral Domain Features for Static Hand Gesture Recognition. In *Proc. of European Signal Processing Conference*. EUSIPCO, 1–5.
[5] B. Alwaely and C. Abhayaratne. 2020. AGSF: Adaptive Graph Formulation and Hand-Crafted Graph Spectral Features for Shape Representation. *IEEE Access* 8, 1 (2020), 1–13.
[6] M. A. Aowal, A. S. Zaman, S. M. Rahman, and D. Hatzinakos. 2014. Static hand gesture recognition using discriminative 2D Zernike moments. In *in TENCON Region 10 Conference*. IEEE, IEEE, 1–5.
[7] M. Asad and C. Abhayaratne. 2013. Kinect depth stream pre-processing for hand gesture recognition. In *Proc. of International Conference on Image Processing*. IEEE, 3735–3739. https://doi.org/10.1109/ICIP.2013.6738770
[8] C. Aslan, A. Erdem, E. Erdem, and S. Tari. 2008. Disconnected skeleton: Shape at its absolute scale. *in IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 12 (2008), 2188–2203.
[9] V. Ayzenberg and S.F. Lourenco. 2019. Skeletal descriptions of shape provide unique perceptual information for object recognition. *Scientific Reports* 9 (2019), 9359.
[10] X. Bai, L. Latecki, and W. Liu. 2007. Skeleton pruning by contour partitioning with discrete curve evolution. *in IEEE Transactions on pattern analysis and machine intelligence* 29, 3 (2007).
[11] X. Bai, W. Liu, and Z. Tu. 2009. Integrating contour and skeleton for shape classification. In *Proc. of international conference on computer vision workshops, ICCV*. IEEE, 360–367.
[12] N. Baker and P. J. Kellman. 2018. Abstract shape representation in human visual perception. *Journal of Experimental Psychology: General* 147, 9 (2018), 1295.
[13] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak. 2011. A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures. *Research Letters in the Information and Mathematical Sciences* 15 (2011), 12–20.
[14] S. L. Brincat and C. E. Connor. 2004. Underlying principles of visual shape selectivity in posterior inferotemporal cortex. *Nature neuroscience* 7, 8 (2004), 880.
[15] M. Carcassoni and E. Hancock. 2002. Alignment using Spectral Clusters.. In *Proc. of British Machine Vision Conference (BMVC)*. 1–10.
[16] A. Chaouch, M.and Verroust-Blondet. 2007. A new descriptor for 2D depth image indexing and 3D model retrieval. In *Proc. of International Conference on Image Processing*, Vol. 6. IEEE, VI–373.
[17] X. Chen, A. Golovinskiy, and T. Funkhouser. 2009. A benchmark for 3D mesh segmentation. *in ACM Transactions on graphics (tog)* 28 (2009), 73.
[18] C. E. Connor, S. L. Brincat, and A. Pasupathy. 2007. Transformation of shape information in the ventral pathway. *Current opinion in neurobiology* 17, 2 (2007), 140–147.
[19] T. Cour, P.n Srinivasan, and J. Shi. 2007. Balanced graph matching. *Advances in Neural Information Processing Systems* 19 (2007), 313.
[20] G. E. da Silva and A. R. Backes. 2018. Characterizing 3D shapes: a complex network-based approach. In *Proc. of European Signal Processing Conference (EUSIPCO)*. 608–612.
[21] M. Demirci, A. Shokoufandeh, and S. Dickinson. 2009. Skeletal shape abstraction from examples. *in IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 5 (2009), 944–952.
[22] E. Dibra, S. Melchior, A. Balkis, T. Wolf, C. Oztireli, and M. Gross. 2018. Monocular RGB hand pose inference from unsupervised refinable nets. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1075–1085.
[23] F. Emmert-Streib, M. Dehmer, and Y. Shi. 2016. Fifty years of graph matching, network alignment and network comparison. *Information Sciences* 346 (2016), 180–197.
[24] P. Felzenszwalb and D. Huttenlocher. 2004. Efficient graph-based image segmentation. *journal of computer vision* 59, 2 (2004), 167–181.
[25] M. Fiedler. 1975. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal* 25, 4 (1975), 619–633.

[26] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok. 2016. A comprehensive performance evaluation of 3D local feature descriptors. *International Journal of Computer Vision* 116, 1 (2016), 66–89.
[27] Z. Han, Z. Liu, C. Vong, Y. Liu, S. Bu, J. Han, and C. Chen. 2017. BoSCC: bag of spatial context correlations for spatially enhanced 3D shape representation. *in IEEE Transactions on Image Processing* 26, 8 (2017), 3707–3720.
[28] R. Horaud and H. Sossa. 1995. Polyhedral object recognition by indexing. *Pattern recognition* 28, 12 (1995), 1855–1870.
[29] H. Jomma and A. Hussein. 2016. Circle Views Signature: A Novel Shape Representation for Shape Recognition and Retrieval. *Canadian Journal of Electrical and Computer Engineering* 39, 4 (2016), 274–282.
[30] L. Latecki, R. Lakamper, and T. Eckhardt. 2000. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc.of the Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. IEEE, 424–429.
[31] Guillaume Lavoué. 2012. Combination of bag-of-words descriptors for robust partial shape retrieval. *The Visual Computer* 28, 9 (2012), 931–942.
[32] M. Leordeanu and M. Hebert. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Proc. of International Conference on Computer Vision (ICCV) Volume 1*, Vol. 2. 1482–1489.
[33] S. Li, H. Lu, and X. Shao. 2014. Human body segmentation via data-driven graph cut. *IEEE transactions on cybernetics* 44, 11 (2014), 2099–2108.
[34] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. 2018. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*. 820–830.
[35] J. Liu, S. Song, C. Liu, Y. Li, and Y. Hu. 2020. A Benchmark Dataset and Comparison Study for Multi-modal Human Action Analytics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 2 (2020), 1–24.
[36] L. Luo, C. Shen, X. Liu, and C. Zhang. 2015. A computational model of the short-cut rule for 2D shape decomposition. *in IEEE Transactions on Image Processing* 24, 1 (2015), 273–283.
[37] X. Luo, F. Wong, and H. Hu. 2020. FIN: Feature Integrated Network for Object Detection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 2 (2020), 1–18.
[38] T. Martinetz and K. Schulten. 1991. A neural-gas network learns topologies. *Artificial Neural Networks* (January 1991), 397–402.
[39] M. Masoumi and A. Hamza. 2017. Spectral Shape Classification: A Deep Learning Approach. *Journal of Visual Communication and Image Representation* (2017).
[40] D. Maturana and S. Scherer. 2015. Voxnet: A 3D convolutional neural network for real-time object recognition. In *Proc. of international conference on Intelligent Robots and Systems (IROS),*. IEEE, 922–928.
[41] L. Minto and P. Zanuttigh. 2015. Exploiting silhouette descriptors and synthetic data for hand gesture recognition. *The Eurographics Association* (2015).
[42] S. O. Murray, B. A. Olshausen, and D. L. Woods. 2003. Processing shape, motion and three-dimensional shape-from-motion in the human cortex. *Cerebral cortex* 13, 5 (2003), 508–516.
[43] L. Nanni, S. Brahnam, and A. Lumini. 2012. Local phase quantization descriptor for improving shape retrieval/classification. *Pattern Recognition Letters* 33, 16 (2012), 2254–2260.
[44] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*. 5099–5108.
[45] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. 2013. Robust Part-Based Hand Gesture Recognition Using Kinect Sensor. *in IEEE Transactions on Multimedia* 15, 5 (Aug 2013), 1110–1120. https://doi.org/10.1109/TMM.2013.2246148
[46] I. Reppa and E. C. Leek. 2019. Surface diagnosticity predicts the high-level representation of regular and irregular object shape in human vision. *Attention, Perception, & Psychophysics* (2019), 1–20.
[47] P. Roberto, F. Emanuele, Z. Primo, M. Adriano, L. Jelena, and P. Marina. 2019. Design, Large-Scale Usage Testing, and Important Metrics for Augmented Reality Gaming Applications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 2 (2019), 1–18.
[48] R. B. Rusu, N. Blodow, and M. Beetz. 2009. Fast point feature histograms (FPFH) for 3D registration. In *Proc. in the International conference on robotics and automation*. IEEE, 3212–3217.
[49] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. 2008. Aligning point cloud views using persistent feature histograms. In *Proc. in the International Conference on Intelligent Robots and Systems*. IEEE, 3384–3391.
[50] T. Sebastian, P. Klein, and B. Kimia. 2004. Recognition of shapes by editing their shock graphs. *in IEEE Transactions on pattern analysis and machine intelligence* 26, 5 (2004), 550–571.
[51] A. Shamaie and A. Sutherland. 2001. Graph-based matching of occluded hand gestures. In *Proc. of Applied Imagery Pattern Recognition Workshop,*. 67–73. https://doi.org/10.1109/AIPR.2001.991205
[52] S. Shapiro and J. Brady. 1992. Feature-based correspondence: an eigenvector approach. *Image and vision computing* 10, 5 (1992), 283–288.
[53] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *in IEEE Signal Processing*

Magazine 30, 3 (May 2013), 83–98. https://doi.org/10.1109/MSP.2012.2235192

[54] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson. 2008. Retrieving articulated 3-D models using medial surfaces. *Machine vision and applications* 19, 4 (2008), 261–275.

[55] A. Sinha, J. Bai, and K. Ramani. 2016. Deep learning 3D shape surfaces using geometry images. In *European Conference on Computer Vision*. Springer, 223–240.

[56] H. Su, S. Maji, E. Kalogerakis, and E. Miller. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *Proc. of international conference on computer vision*. 945–953.

[57] B. Taati and Mi. Greenspan. 2011. Local shape descriptor selection for object recognition in range data. *Computer Vision and Image Understanding* 115, 5 (2011), 681–694.

[58] Y. S. Tan, K. M. Lim, C. Tee, C. P. Lee, and C. Y. Low. 2020. Convolutional neural network with spatial pyramid pooling for hand gesture recognition. *Neural Computing and Applications* (2020), 1–13.

[59] S. Umeyama. 1988. An eigendecomposition approach to weighted graph matching problems. *in IEEE Transactions on Pattern Analysis and Machine Intelligence* 10, 5 (Sept 1988), 695–703. https://doi.org/10.1109/34.6778

[60] D. Vranic and D. Saupe. 2002. Description of 3D-shape using a complex function on the sphere. In *Proc. of International Conference on Multimedia and Expo*, Vol. 1. IEEE, 177–180.

[61] B. Wang, W. Shen, W. Liu, and X. Bai. 2010. Shape classification using tree-unions. In *Proc. of Pattern Recognition*. (ICPR), 983–986.

[62] C. Wang, Z. Liu, and S. C. Chan. 2015. Superpixel-based hand gesture recognition with Kinect depth camera. *in IEEE Transactions on Multimedia* 17, 1 (2015), 29–39.

[63] X. Wang, Y. Tian, X. Zhao, T. Yang, J. Gelernter, J. Wang, G. Cheng, and W. Hu. 2020. Improving Multiperson Pose Estimation by Mask-aware Deep Reinforcement Learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 3 (2020), 1–18.

[64] R. Watt and D. Andrews. 1982. Contour curvature analysis: hyperacuities in the discrimination of detailed shape. *Vision research* 22, 4 (1982), 449–460.

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

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

[67] Z. Yasseen, A. Verroust, and A. Nasri. 2016. Shape matching by part alignment using extended chordal axis transform. *Pattern Recognition* 57 (2016), 115–135.

[68] C. Zahn and R. Roskies. 1972. Fourier descriptors for plane closed curves. *in IEEE Transactions on computers* 100, 3 (1972), 269–281.

[69] Y. Zheng, B. Guo, Y. Yan, and W. He. 2019. O2O Method for Fast 2D Shape Retrieval. *IEEE Transactions on Image Processing* 28, 11 (2019), 5366–5378.

[70] S. Zhi, Y. Liu, X. Li, and Y. Guo. 2018. Toward real-time 3D object recognition: A lightweight volumetric CNN framework using multitask learning. *Computers & Graphics* 71 (2018), 199–207.

[71] F. Zhou and F. De La Torre. 2015. Factorized Graph Matching. *in IEEE Transactions on Pattern Analysis and Machine Intelligence* PP, 99 (2015), 1–1. https://doi.org/10.1109/TPAMI.2015.2501802

[72] Y. Zhou and F. Zeng. 2017. 2D compressive sensing and multi-feature fusion for effective 3D shape retrieval. *Information Sciences* 409 (2017), 101–120.