Graph Spectral Feature Learning for Shape Representation

Basheer Alwaely

Supervisor:

Dr. Charith Abhayaratne

Thesis submitted in candidature for graduating with a degree of doctor of philosophy from the University of Sheffield, Faculty of Engineering, Department of Electronic and Electrical Engineering, July 26, 2019.

Abstract

Two decades ago, it was difficult to imagine a machine that can auto-detect patterns in images. The state-of-the-art studies have advanced shape recognition, which explores the representation of each modality. However, it remains a challenge for machines to learn shape representations accurately. One way to describe the topology of a shape is to use feature representations. Discriminative features have attracted attention because of their potential in recognising patterns with more accuracy at rapid execution speeds. In this thesis, we first introduce a novel method for in-air arbitrary shape recognition. Particularly, we propose a subset of graph spectral features, which are invariant to rotation, flip, and mirror changes, to classify different shape categories. A new dataset is also introduced for in-air hand-drawn that includes samples of shapes and numbers. This method has resulted in the highest performance with accuracies of 99.56% and 99.44%, for numbers and shapes, respectively, outperforming the existing methods for different datasets. The second contribution involves the development of an adaptive graph connectivity method based on the local details. Such graphs are created to fit the topology of the targeted shapes. A set of spectral graph features are then extracted to capture the local details, improving the state-of-the-art performance by 2% and 9% for two 2D datasets, and 2% and 6% for two 3D datasets. The third contribution focuses on simplifying the shapes by further exploring the local details. A spectral partitioning is applied to understand the geometric structure of each part. This is followed by extracting the spectral features to identify the shapes. The empirical evaluation shows that partitioning of the shapes provides deep geometric details that have demonstrated increasing accuracy levels in different datasets by 1.02%, 5.09%, 2.1%, and 7.89%.

Contents

List of Figures

List of Tables

List of symbols

Γ	The geometric graph Laplacian matrix
γ	Filter parameter
δ	The increment value of the threshold
$\mathcal{E}$	Graph edges
θ	The angle signal
k	Number of samples in dataset
λ	Graph Eigenvalue
$\mathcal{V}$	Graph vertices
Φ	Number of incident vertices
ω	Number of clusters
$\mathcal{L}$	The normalised graph Laplacian matrix
$\mathbf{A}$	Graph adjacency matrix
$\mathbf{D}$	Degree matrix
$\mathbf{d}_i$	Dataset
E	Entropy
F	Features
$\mathcal{G}$	Graph
$\hat{h}$	Filter
$\mathcal{I}$	Graph incidence matrix
l	Label
i, j	Index
K	Number of nearest features
$\mathbf{L}$	The combinatorial Graph Laplacian matrix
$\mathbf{M}$	Modulation matrix
m	Image resizing scaler
N	Original number of observation
n	Number of observation after shape representation

P	Original hand path line
$\hat{P}$	Hand path line after downsampling
$\hat{\mathbf{r}}$	Distance function
S	Matrix representation of a group of graph eigenvectors
T	Threshold
$t_\circ$	Conditional threshold
$\mathbf{U}$	Graph Eigenvector
$\mathbf{u}$	Individual graph Eigenvector
X, Y, Z	Original shape coordinates
x, y, z	Shape coordinates after pre-processing

List of abbreviations

2D	Two dimensional shape
3D	Three dimensional shape
BoF	Bag-of-Features
CAT	Chordal Axis Transform
CboFHKS	Compact bag-of-features Heat Kernel Signatures
CNN	Convolution Neural Networks
CT	Classification Tree
DPM-BCF	Depth Project Map based Bag of Contour Fragments
DSP	Discrete Signal Processing
DTW	Dynamic Time Warping
DZM	Dimensional Zernike moments
ETU10	Economics and Technology University dataset
FD	Fourier Descriptors
FEMD	Finger Earth Mover's Distance
FS	Finger Segmentation
GA	A combination of Graph and Appearance features
GNG	Growing Neural Gas
GSF	Graph Signal Features
GSP	Graph Signal Processing
H3DF	Histogram of 3D Facets
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
KNN	Nearest Neighbour classifier
LCDT	Local Curvature with Distance Transform
LDA	Linear Discriminant Analysis
MDLA	Multi-view depth line
MST	Maximum Spanning Tree

NN	Neural Network
Proc.	Proceeding
QDA	Quadratic Discriminant Analysis
RF	Random Forest
sEMG	Surface electromyography
SGWT	Spectral Graph Wavelet Transform
SHREC	Shape Retrieval Contest dataset
SIFT	Scale Invariant Feature Transform
SP-EMD	Super-Pixel Earth Mover's Distance
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TM	Template Matching
ZM	Zernike Moments

Declaration

This thesis is the result of my own work, ideas, experiments and has not previously been submitted or accepted for any degree other than Doctor of Philosophy of the University of Sheffield. This thesis includes materials, which have been appeared in a published journal and conference papers.

Acknowledgements

I would like to take this opportunity to express my gratitude to my supervisors Dr. Charith Abhayaratne who provided insight and expertise that greatly assisted the research. I am especially indebted to my supervisor for sharing his pearls of wisdom and experience during my doctoral studies.

Special thanks to the Ministry of Higher Education and Scientific Research (Iraq) for their scholarship.

Also, I am sincerely grateful to my family for their love and support in whatever I pursue.

Chapter 1

Introduction

Shape recognition has received great attention in the field of computer vision. The detection of shape appearance, part-structure, occlusion, articulation, and local details have played an important role in shape classification. The representation of such characteristics is particularly significant when it comes to distinguishing highly similar shapes. This is often the case in existing shape datasets, which consist of similar and complex shapes, leading to ambiguity in shape recognition.

This chapter presents the main motivation behind, and challenges involved in shape identification. This chapter is organised as follows, Section 1.1 presents the motivation behind studying shape recognition and crystallises the main research question. The key challenges facing shape recognition techniques are highlighted in Section 1.2. Section 1.3 highlights the contributions of this thesis. A list of publications based on the contents of the thesis are listed in Section 1.4. Section 1.5 outlines the structure of the thesis.

1.1 Motivation

Over the last decade, there have been significant contributions to the shape representation literature. In particular, shape recognition has benefited from great efforts in the computer vision community, resulting in its remarkable evolution [4,5]. This is because of its extensive applications in many fields, which have a direct impact on everyday life [6–9].

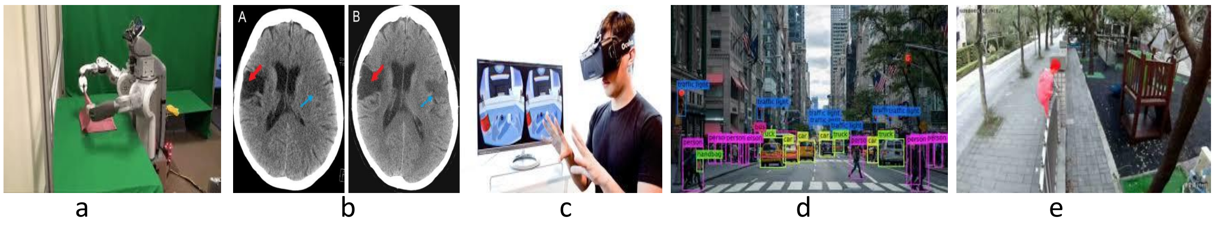


Figure 1.1: Some examples of computer vision applications; (a) Robotic `http://www.luigifreda.com/`; (b) Virtual reality `https://www.blippar.com/`; (c) Medical imaging `https://en.wikipedia.org/`; (d) Object detection and recognition `https://towardsdatascience.com/`; (e) Security `https://www.vision-systems.com/`.

Examples of these applications can be found in the industrial applications, such as using robots for human assistance (Figure. 1.1a), and medical imaging for accurate and fast diagnosis (Figure. 1.1b). Moreover, the recent development of computer vision leads to a great improvement in the virtual reality and video gaming (Figure. 1.1c) especially with the emerging of deep learning. From detection, tracking, segmentation, and recognition, all of these processes can be performed accurately using computer vision, which could be useful in a wide range of applications such as human detection, and self-driving cars (Figure. 1.1d). As a result, computer vision provides an efficient way to improve the security in homes (Figure. 1.1e), airports, and fingerprint identification.

Humans possess one of the most complex visual systems, and have an extraordinary ability to capture a variety of shape details. In order to understand human perception, sociologists have shown that human sight detects different complex objects, based on visual acuity (*i.e.*, ability to see fine details), depth, colour, movement, and contrast. In addition to these features, many researchers have highlighted the importance of surface curvature details for shape understanding [10–13]. This thesis therefore aims to analyse shapes based on the global and local details of their surfaces, taking into account a variety of structures.

From an engineering perspective, shape recognition has been explored using different methods, including via deep learning [14], graph theory [15], feature extraction [16], view similarities [17], and model similarities [18]. Specifically, graphs have been widely used in the field of computer vision, and have been demonstrated to be powerful tools with many applications, such as pattern recognition, segmentation, and clustering [19–22]. Graphs are applied in shape recognition in two ways. The first method, which has been used in many studies, utilises approximate similarity measurement methods through the adjacency matrices of two samples [15, 23–29]. In the second method, a bipartite graph matching is performed based on the node domain to find the optimal fit map between two samples [30–32]. However, graph-based methods are limited in their performance to address the problems of accurate recognition, rotation changes, and time complexity. Graph-based methods are rotation invariant but time-consuming. In contrast, feature-based methods are fast but sensitive to angle changes.

Therefore, in this thesis, we use new hybrid methods that combine graph and features based approaches. Particularly, we explore the utility of graph spectral features for shape recognition to achieve real-time and rotation invariant shape representation, rather than graph matching in the node domain. The main research question here is to develop and test the ability of the graph spectral domain features to provide optimal shape recognition methods. To achieve this goal, we need,

1. To explore the ability of the graph spectral features to capture global details.
2. To investigate the reliability of the graph spectral features to reveal fine details.
3. To examine the efficiency of the graph spectral features representation for real-time performance and rotational changes.
4. To evaluate the performance of the graph spectral features of shape partitioning, in order to have a deep understanding of the shape topology.

These objectives outline the research methodology. In the following subsections, we will highlight the main difficulties in shape recognition methods.

1.2 Challenges

Identifying shapes via computer is a difficult task, and there are many problems that need to be addressed when designing an optimal recognition system [33]. These issues may include shape representation, scaling differences, fine details detection, the distinction among high similarity objects, and performance complexity. The major challenges for shape recognition are summarised below:

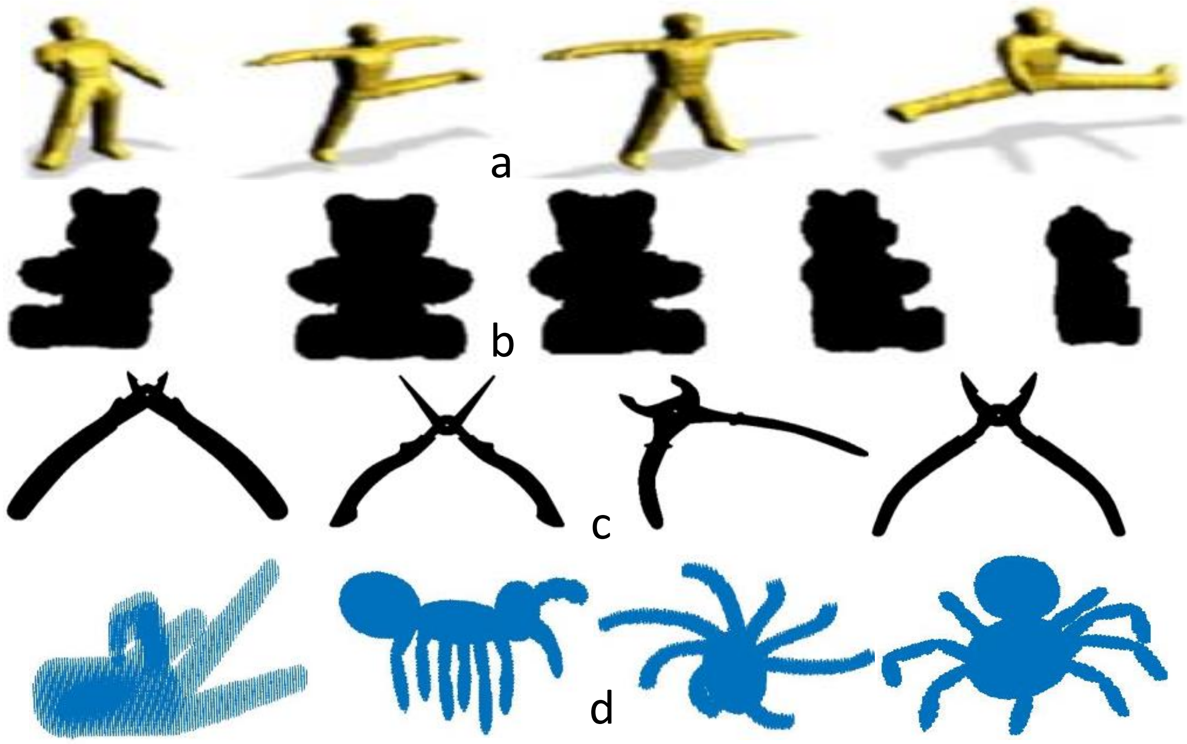


Figure 1.2: Challenging shapes for recognition methods. The difficulty varies depending on the type of structure, articulation, rotational invariant, and the similarity among different categories.

1. **Global shape detection**

   One of the major challenges to an accurate and robust shape recognition system is to perfectly represent the geometric structure of the shape. This has previously been implemented through a set of descriptors, view similarities, model representations, and deep learning techniques. However, the aim to identify increasingly complex objects makes the recognition process difficult to implement. For example, Figure. 1.2(row a) [3] shows the different geometric details belong to one class. In this scenario, more features are needed to provide an optimal description of the shape. Similarly, more data are required to train the deep learning approaches in order to achieve an acceptable performance. An efficient recognition method should perfectly characterise the geometric details of the structure.

2. **Angle changes**

   Another challenging issue in shape recognition studies is sensitivity to rotational changes. The existing datasets display objects from each class in different views, as shown in Figure. 1.2(row b). Moreover, some objects are provided in different sizes, which adds further difficulties to the recognition process. An optimal classification system must be able to recognise the same object from different views and at different sizes.

3. **Local details detection**

   Recognising and interpreting small variations in objects that are distinctly similar in their global structure is a crucial factor in distinguishing between different patterns. The human eye can easily discriminate patterns in images. However, a manual classification process for these patterns is a challenging task for researchers. For example, objects may be classified in the same class, even though they have different types of geometric structures, as shown in Figure. 1.2(row c). This is because the similarities between the patterns are numerous, in terms of global structure, while relatively few differences are noticed. As a result, the interpretation of small details is an important factor for distinguishing between different shapes. This highlights the importance of exploiting differences, in terms of protrusions and other fine details, as well as the global shape.

4. **Deep understating of the topology**

   More complex situations occur in recognising shapes that have similar geometric details and the same number of limbs, as shown in Figure. 1.2(row d). The ability to classify such shapes usually depends on deep learning approaches. This is because it is difficult to achieve an effective description simply through features or models. Therefore, any feature representation of shapes should provide accurate deep information of local and global details.

1.3 Key contributions

The main contributions of this thesis are:

1. **The proposal of a new rotation and flip-invariant method for in-air hand drawn**

shape recognition with real-time operation

A new shape recognition method is proposed, based on the global details for in-air hand-drawn shape recognition. **Chapter 3** addresses the problem of matching shapes by representing the structures as a graphical form and extracting the features based on the graph spectral domain. The first contribution of this thesis includes pre-processing for converting the hand path movement, captured via Kinect, into a fully connected graph, followed by analysis of the eigenvectors of the normalised Laplacian matrix to form feature vectors. This method uses the eigenvector corresponding to the lowest eigenvalue to formulate the feature vectors, as it captures the global details of the structure. The proposed method has achieved average accuracy rates of 99.56% and 99.44%, for numbers and shapes, respectively, outperforming the existing methods. It also adds the benefits of real-time operation and invariance to rotation and flipping, making the recognition system robust to different writing and drawing variations.

2. **An in-air hand-drawn number and shape dataset**

   A new data set of in-air hand-written numbers and shapes samples is collected and uploaded for public use. More details about this dataset are presented in **Chapter 3**.

3. **A new method for graph formulation with adaptive connectivity to represent shapes by capturing their local and global characteristics**

   A new method is presented, based on local details for 2D shape matching, which is exploited for 3D shape matching, in **Chapter 4**. An adaptive graph is created to accurately contain the shape architecture, including its fine details in such a way that shape protuberances are highlighted and interpreted via node connectivity. We formulate an adaptive graph connectivity based on a certain threshold associated with the shape's structure. Such generated graphs show the arrangement of prominent parts along the border of the shape in great detail. This is followed by a feature extraction process using the graph spectral and node domains. The experimental evaluations show that the containment of local details reduces the error rate in the matching process and improves the accuracy level of the ETU10, Tool, SHREC2010, and 3D shape benchmark datasets by 2%, 9%, 2% and 6% respectively.

4. **A shape partitioning method for matching purposes**

   With more complex shapes being identified, it has become necessary to obtain a deep understanding of their details. This can be achieved by simplifying the shape, along with analysing the global and local details of the structure. **Chapter 5** therefore presents a new method, based on local and global details for 3D shape recognition, which we also applied to 2D shape and hand gesture recognition. Local and global details of the shapes are detected using the graph spectral domain features. A fully connected graph is generated over the silhouette of the shape to describe their global structures. We also introduce a fully automatic, divisive, hierarchical clustering method based on a skeleton representation, for matching purposes. In practice, we demonstrate the ability of the Fiedler vector to provide a stable partitioning. The evaluation process manifests the performance of the proposed method compared to state-of-the-art studies, by increments of 1.02%, 5.09%, 2.1%, 7.89% for four datasets.

1.4 Publications

Part of the work in this thesis has been published in the following journal and conference papers.

Journal papers:

1. B. Alwaely and C. Abhayaratne. “ Graph Spectral Features for In-Air Hand-Drawn Number and Shape Recognition” submitted toward IEEE Access, October, 2019.

2. B. Alwaely and C. Abhayaratne. “ Spectral Features of Adaptively Connected Graphs for 2D and 3D Shape Representation ” submitted toward IEEE transaction on image processing, October, 2019.

3. B. Alwaely and C. Abhayaratne. “ Shape Simplifying Based on Graph Spectral Domain ” to be submitted toward IEEE transaction on Systems, Man, and Cybernetics, 2019.

Conference papers:

1. B. Alwaely and C. Abhayaratne. “ Graph Spectral Domain Feature Representation for in-Air Drawn Number Recognition” Proc. of European Signal Processing Conference (EUSIPCO), 2017, pp. 370-374.

2. B. Alwaely and C. Abhayaratne. “ Graph Spectral Domain Shape Representation” Proc. of European Signal Processing Conference (EUSIPCO), 2018, pp. 603-607.

3. B. Alwaely and C. Abhayaratne. “ Adaptative Graph Formulation for 3D Shape Recognition” Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE, 2019, pp. 1947-1951.

4. B. Alwaely and C. Abhayaratne. “ Graph Spectral Domain Features for Static Hand Gesture” Proc. of European Signal Processing Conference (EUSIPCO), 2019, pp. xxx-xxx.

1.5 Outline

The overall structure of the thesis takes the form of six chapters, which are listed as follows:

Chapter 2 provides a detailed description of the existing works on shape recognition. Section 2.1 gives an insight about different type of structures. Section 2.2 illustrates graph fundamentals including the graph theory, graph spectral domain, graph applications. Section 2.3 explores the existing works of the in-air hand written shape recognition. 2D/3D shape recognition studies are explained in Section 2.4. Section 2.5 reviews the existing work of the static hand gesture recognition. This chapter will be summarised in Section 2.6.

Chapter 3 introduces a novel method for in-air hand drawn number and shape recognition based on a graph spectral features representation. Section 3.1 provides a general introduction to the proposed methods. Section 3.2 presents the proposed graph spectral features in the context of an in-air hand drawn number recognition system. Section 3.3 provides the performance evaluation, followed by a summary of the work in Section 3.4.

In **Chapter 4** a new method is proposed, based on the graph spectral domain for 2D/3D shape recognition. Section 4.1 introduces the proposed adaptive graph generation. A comprehensive explanation of the graph concepts, the graph model based on adaptive connectivity and graph spectral features are shown in Section 4.2. Then, Section 4.3 evaluates the proposed system based on different classifiers and data-sets. Finally, the work is summarised in Section 4.4.

Chapter 5 presents a new method for hand gesture recognition, which we exploited for 2D/3D shape matching based on shape silhouette and skeleton representation. See Section 5.1 for an introduction to the use of the graph spectral domain to detect the global shape and the utility of the graph for simplifying the structure. In Section 5.2 the proposed method is illustrated in detail, including the graph concepts, the silhouette and skeleton representation, and the proposed features. A performance evaluation of the proposed method, using publicly available hand gestures databases and shape datasets is presented in Section 5.3 including testing different classifiers and parameters. A cross validation performance of all the proposed method will be shown in Section 6.2. This work is summarised in Section 5.4.

Chapter 6 brings together the contents of the thesis and outlines potential future directions for shape recognition studies.

Chapter 2

Background and related work

2.1 Introduction

As mentioned above, in this chapter, we provide a detailed background on the graph theory and review the most relevant works pertinent to shape recognition. In general, shapes can be seen from three different perspectives (low, medium, and high level detailed shapes). For example, Figure. 2.1(row1) shows a set of shapes that can be rendered using a skeleton representation. In other words, they can be drawn using a single line, from the starting point to the endpoint. We call this type of shape a low-level structure and these can be found in applications related to human hand movements. Another type of structure, which can not be represented by a single line, is called a medium-level structure. In such structures, the fine details of a silhouette representation are needed in order to understand the shape, whereas a skeleton representation might cause similarity among different classes as can be seen in Figure. 2.1(row2). These shapes are often found in 2D shape recognition applications. High-level structures contain more complex details in their topologies, and are usually 3D shapes, such as 3D point cloud shapes and mesh structures. Figure. 2.1(row3) illustrates an example of such structures. From this, it can be observed that even the skeleton and silhouette representations of these structures have conceptual similarities.

Therefore, this study aimed to develop an ideal method for the three related structures in shape understanding. First, we explore the differentiation of shapes in terms of global details to recognise low-level structures. Since these structures can be represented by a single line, the

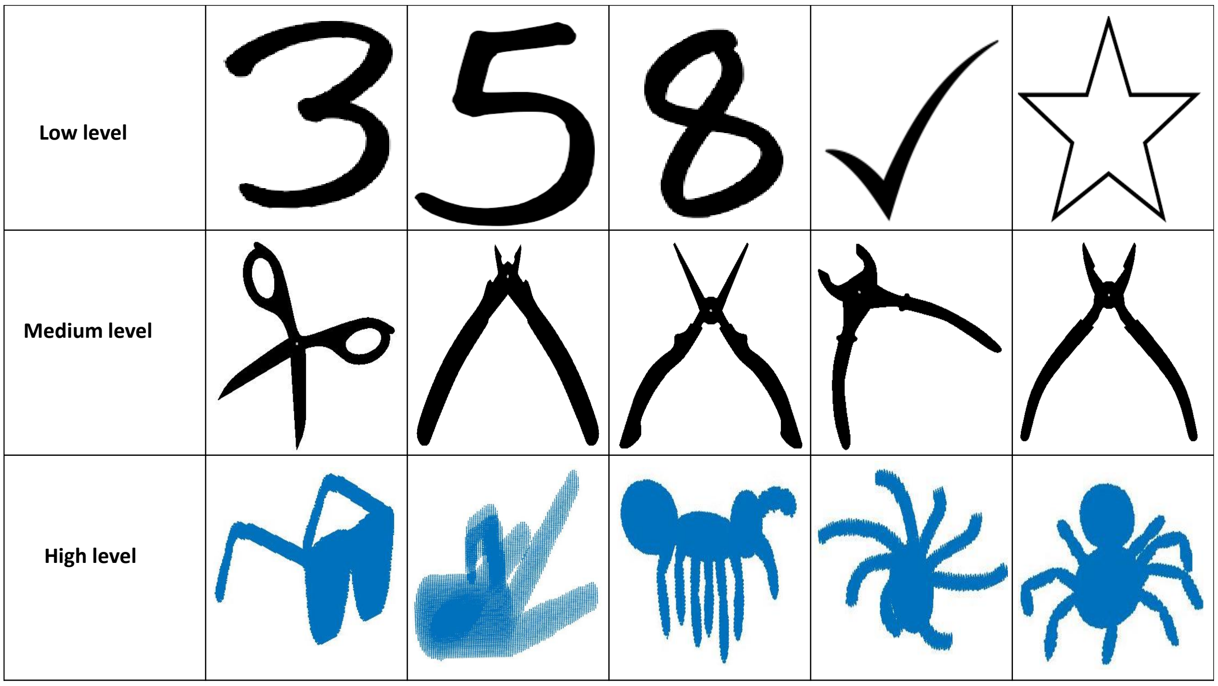


Figure 2.1: Different types of shapes according to its structure. Based on their complexity, we divided them into three levels, low, medium and high levels structure.

solution to this problem is to be able to detect patterns with high accuracy level, fast execution, and irrespective of rotation. Second, we exploit distinctions in shapes, in terms of protrusions and fine details in the shape contour as well as the global shape description. Finally, we investigate the ability of graph spectral domain bases for shape partitioning in order to have a deeper understanding of each part of the shape. This partitioning process is able to provide a stable fragmentation process for different shapes.

This chapter is divided into six sections, which explains how we approached achieving these objectives. Initially we provide preliminary information on graph theory, spectral graph theory, and graph signal process operations in Section 2.2. Then, we review relevant works in the field of in-air hand drawn shape recognition in Section 2.3 as an application for low level structures. 2D/3D shapes are considered as an applications of the medium, and high complexity structures, and they have almost the same techniques as will be shown in Section 2.4. In addition, we applied the proposed methods for hand gesture recognition, and so works of hand gesture recognition are discussed in Section 2.5. This chapter is summarised in Section 2.6.

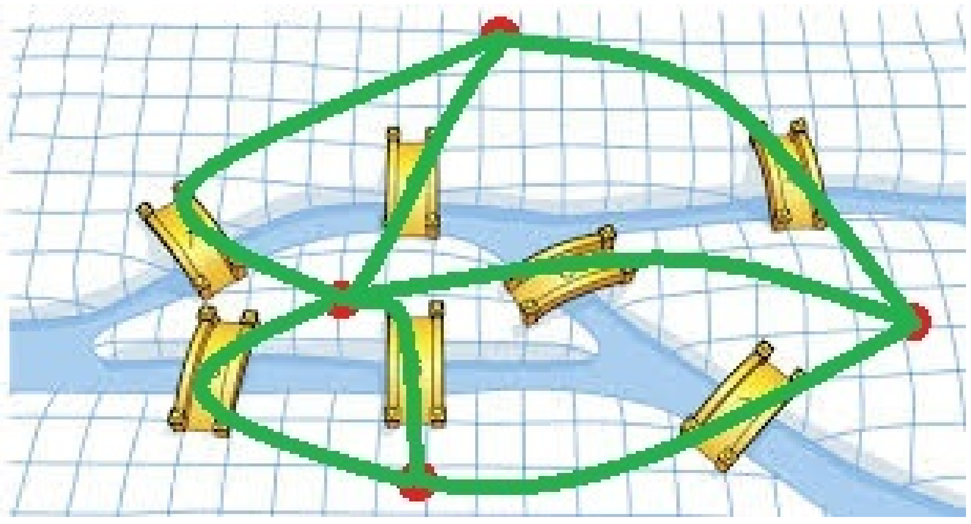


Figure 2.2: Seven bridges of Konigsberg and its graph representation.

2.2 Graph spectral domain concepts

In this section, we cover the areas of graph theory, graph spectral theory, and related applications. Historically, graph theory was invented by Leonhard Euler when he solved the Bridges of Konigsberg city problem (see Figure. 2.2). Three lands were connected to each other with the mainland by seven bridges, and all the lands could be accessed by walking cross all the bridges once, which is known later by the Eulerian graph. Since 1736, the idea of graph theory was created. In the past decade, great attention has been given to graph signal processing [34] due to its benefits in supporting numerical applications, such as social networks, sensors, image processing, partitioning, and transportations [35]. Also, significant effort has been devoted to expanding and applying traditional basic signal processing methods to graphs such as down sampling, Fourier transforms, compression, and wavelet [35, 36] as will be shown in the following subsection.

2.2.1 Graph theory

The graph is a mathematical representation of the structure data (*e.g.*, regular and irregular data). A graph in this context comprises nodes, which are connected by edges. Different graph properties and its related applications based on the node domain are shown as follows,

Direct and undirected graphs are two types of graph edges form. Undirected graph means that the same edge can be used to go and come back between vertices. The vitality of the undirected graph is in the adjacency matrix, which is symmetric and results in real eigenvalues and eigenvectors, whereas, a direct graph may form ill-posed situated in the graph eigenvectors matrix.

Connected and unconnected graphs are based on the available edges between the nodes. A graph is called connected when its vertices have, at least, one edge between them to connect as one group, while the graph is called unconnected graph if there is, at least, one node is not connected to other nodes. Graph connectivity is an important property in applications related to cycles [37] or pathfinder [38]. The connectivity could be also detected by the graph eigenvalues as will be shown later in Chapter 4, where the number of zero eigenvalues refers to the number of connected groups.

2.2.2 Graph spectral analysis

Let $\mathcal{G}$ be an undirected graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$, where $\mathcal{V}$ is the set of n vertices, $\mathcal{E}$ is the set of edges and $\mathbf{A}$ is the adjacency matrix with edge weights. We consider $\mathcal{G}$ as a fully connected graph, which means each vertex has $(n-1)$ connected edges. We define the weight, $\mathbf{A}_{(i,j)}$ corresponding to an edge, $e_{(i,j)}$ connecting vertices i and j is as follows:

$$\mathbf{A}_{i,j} = \frac{|e_{(i,j)}|}{\frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}|e_{(i,j)}|}, \tag{2.1}$$

$e_{(i,j)}$ is the Euclidean distance between the vertices, i and j, normalised with the average edge length of the nodes. There is another alternative way of interpreting the relationship between nodes by considering the value equal to one to each pair of connected nodes and zeros otherwise. This matrix representation is defined as the incidence matrix ($\mathcal{I}$) as shown in Eq. (2.2).

$$\mathcal{I} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected;} \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

The combinatorial graph Laplacian matrix or the non-normalised version, $\mathbf{L}$, is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{2.3}$$

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

$$\mathbf{D}_{(i,i)} = \sum_{j=0}^{n-1} \mathbf{A}_{(i,j)}, \qquad i = 0, 1, ..., n-1. \tag{2.4}$$

We also consider the symmetric normalised Laplacian matrix, ($\mathcal{L}$), which is computed as follows:

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}, \tag{2.5}$$

and the geometric Laplacian matrix (Γ), is computed as follows:

$$\Gamma = \mathbf{D}^{-1}\mathbf{A}. \tag{2.6}$$

Since, the Laplacian matrices, $\mathbf{L}$, $\mathcal{L}$, and Γ are symmetric positive semidefinite matrices, from spectral projection theorem, there exists a real unitary matrix, $\mathbf{U}$, that digonalises $\mathcal{L}$, such that $\mathbf{U}^t\mathcal{L}\mathbf{U} = \Lambda = diag\{\lambda_\ell\}$ is a non-negative diagonal matrix [39], leading to an eigenvalue decomposition of $\mathcal{L}$ matrix as follows:

$$\mathcal{L} = \mathbf{U}_{\mathcal{L}}{}^t \lambda \mathbf{U}_{\mathcal{L}} = \sum_{\ell=0}^{n-1} \lambda_\ell \mathbf{u}_\ell \mathbf{u}_\ell^t, \tag{2.7}$$

where $\mathbf{u}_\ell$, the column vectors of $\mathbf{U}$, are the set of orthonormal eigenvectors of $\mathcal{L}$ with corresponding eigenvalues, $0 = \lambda_0 < \lambda_1 \leq \lambda_2 ... \leq \lambda_{n-1} = \lambda_{max}$ [35].

An example of a random undirected graph can be seen in Figure. 2.3, with its corresponding adjacency matrix Eq. (2.8), incidence matrix Eq. (2.9), combinatorial eigenvectors Eq. (2.10), normalised eigenvectors Eq. (2.11), and the geometric eigenvectors Eq. (2.12). In this example, we can see the negative values of the second eigenvector are associated with the poorly connected nodes, while the positive values are associated with the strong connectivity.

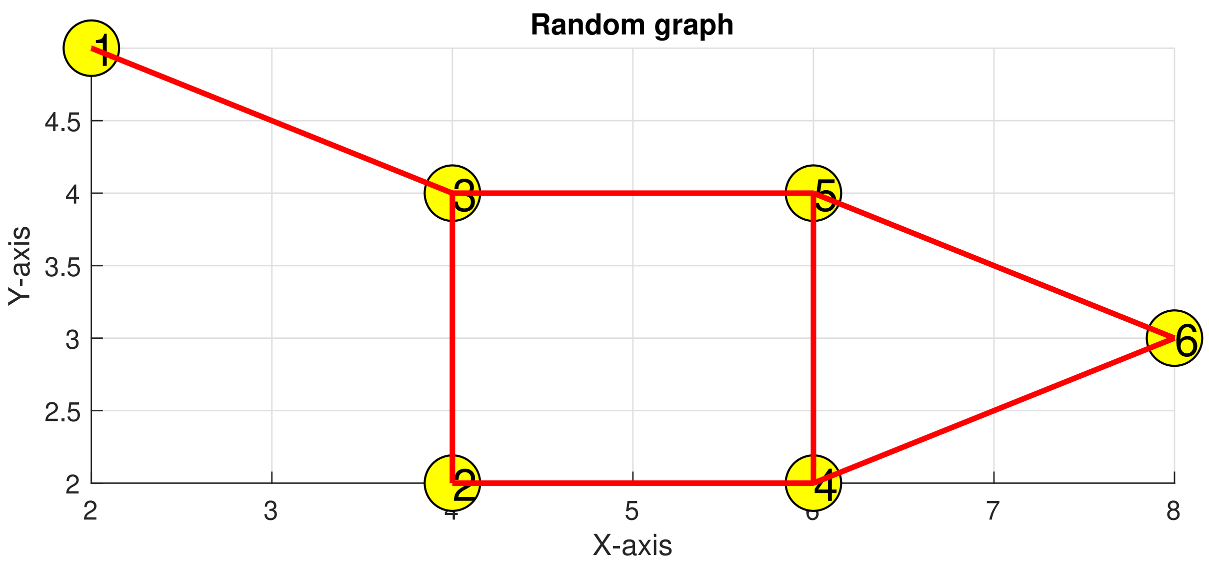


Figure 2.3: Example of a random six-node graph with it is connectivity.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0.0127 & 0 & 0 & 0 \\ 0 & 0 & 0.0113 & 0.0113 & 0 & 0 \\ 0.0127 & 0.0113 & 0 & 0 & 0.0113 & 0 \\ 0 & 0.0113 & 0 & 0 & 0.0113 & 0.0127 \\ 0 & 0 & 0.0113 & 0.0113 & 0 & 0.0127 \\ 0 & 0 & 0 & 0.0127 & 0.0127 & 0 \end{bmatrix}, \tag{2.8}$$

$$I = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \tag{2.9}$$

$$\mathbf{U_L} = \begin{bmatrix} 0.4082 & -0.7786 & 0.2942 & 0.3232 & 0.0943 & -0.1649 \\ 0.4082 & 0.0592 & -0.7876 & 0.2299 & -0.2032 & -0.3397 \\ 0.4082 & -0.2472 & -0.1662 & -0.5958 & -0.2431 & 0.5749 \\ 0.4082 & 0.3204 & -0.0312 & 0.3247 & 0.6494 & 0.4500 \\ 0.4082 & 0.2291 & 0.2312 & -0.5548 & 0.3121 & -0.5676 \\ 0.4082 & 0.4171 & 0.4595 & 0.2728 & -0.6095 & 0.0473 \end{bmatrix}, \tag{2.10}$$

$$\mathbf{U}_{\mathcal{L}} = \begin{bmatrix} -0.2757 & 0.5608 & 0.2980 & -0.4300 & -0.5467 & -0.1923 \\ -0.3688 & 0.1185 & -0.7463 & -0.4360 & 0.1426 & 0.2874 \\ -0.4604 & 0.5387 & 0.0648 & 0.6265 & 0.2662 & 0.1738 \\ -0.4604 & -0.3685 & -0.3073 & 0.3235 & -0.3700 & -0.5624 \\ -0.4604 & -0.2286 & 0.4017 & -0.3571 & 0.6007 & -0.2933 \\ -0.3899 & -0.4395 & 0.3071 & 0.0163 & -0.3350 & 0.6693 \end{bmatrix}, \tag{2.11}$$

$$\mathbf{U}_{\Gamma} = \begin{bmatrix} -0.1781 & -0.3907 & -0.2108 & -0.2855 & -0.3732 & -0.1271 \\ -0.3186 & -0.1104 & 0.7060 & -0.3873 & 0.1302 & 0.2542 \\ -0.4966 & -0.6268 & -0.0765 & 0.6948 & 0.3035 & 0.1919 \\ -0.4966 & 0.4288 & 0.3630 & 0.3587 & -0.4218 & -0.6210 \\ -0.4966 & 0.2660 & -0.4745 & -0.3960 & 0.6849 & -0.3239 \\ -0.3562 & 0.4331 & -0.3072 & 0.0153 & -0.3235 & 0.6259 \end{bmatrix}. \tag{2.12}$$

One question that needed to be asked in this study was about which Laplacian matrix should be used. From the literature review, there is no clear evidence concerning which was the optimal version of the graph Laplacian matrix. However, it had been found [35] that both versions have the similar notion of frequency, based on the number of zero crossing in their eigenvectors. In general, researchers have tended to use $\mathbf{L}$ in applications related to image processing [40, 41]. This is because $\mathbf{L}$ provides useful image bases, such as the direct current (DC) value in the eigenvector corresponding to the first eigenvalue which is not the same case for $\mathcal{L}$. Figure. 2.4 shows an example of line graph with 8 nodes, and it is corresponding bases of the $\mathbf{L}$ or $\mathcal{L}$.

The normalised Laplacian matrix was first introduced [20] to minimise the generalised eigen problem. Basically, the combinatorial version optimises the objective based on the number of nodes in each group, while the normalised version optimises the graph relative to the volume of each group. For example, Figure. 2.5 illustrates the behaviour of both versions using different graphs of handwritten numbers (0-9). We can see that the combinatorial version is sensitive to the individual node connectivity, whereas the normalised version provides a general description of the graph. Hence, we can conclude that the normalised version can be used for understanding global shape, while the combinatorial version reveals the local details of the structure.

The graph signal process offers new opportunities for the processing, compression and analysis of spatially non-uniformly sampled data, represented as a graph, by characterising the global structure, based on its eigenvalues and the eigenvectors of the graph Laplacian matrix [35]. The graph eigenvectors (*e.g.*, bases vectors as in a content adaptive transform) provide an efficient representation of the connectivity and the structure of the graph. One of the most highlighted basis is the eigenvector corresponding to the second smallest eigenvalue, which is known as the Fiedler vector [42] or the algebraic connectivity [43]. The sign of Fiedler vector has been explored in analysis tasks, such as, determining the stability of system [44], saliency estimation [45] and image partitioning [20]. Fiedler vector has been proven to be a powerful division tool by splitting the graph into two parts based on its sign and according to the optimisation formula,

$$\lambda_1 = min\left[\frac{\mathbf{U}^t\mathbf{L}\mathbf{U}}{\mathbf{U}^t\mathbf{U}}\right], \tag{2.13}$$

which has been proven in Eq. (2.14) [42], where $\left[\frac{\mathbf{U}^t\mathbf{L}\mathbf{U}}{\mathbf{U}^t\mathbf{U}}\right]$ is a vector and λ_1 represents the

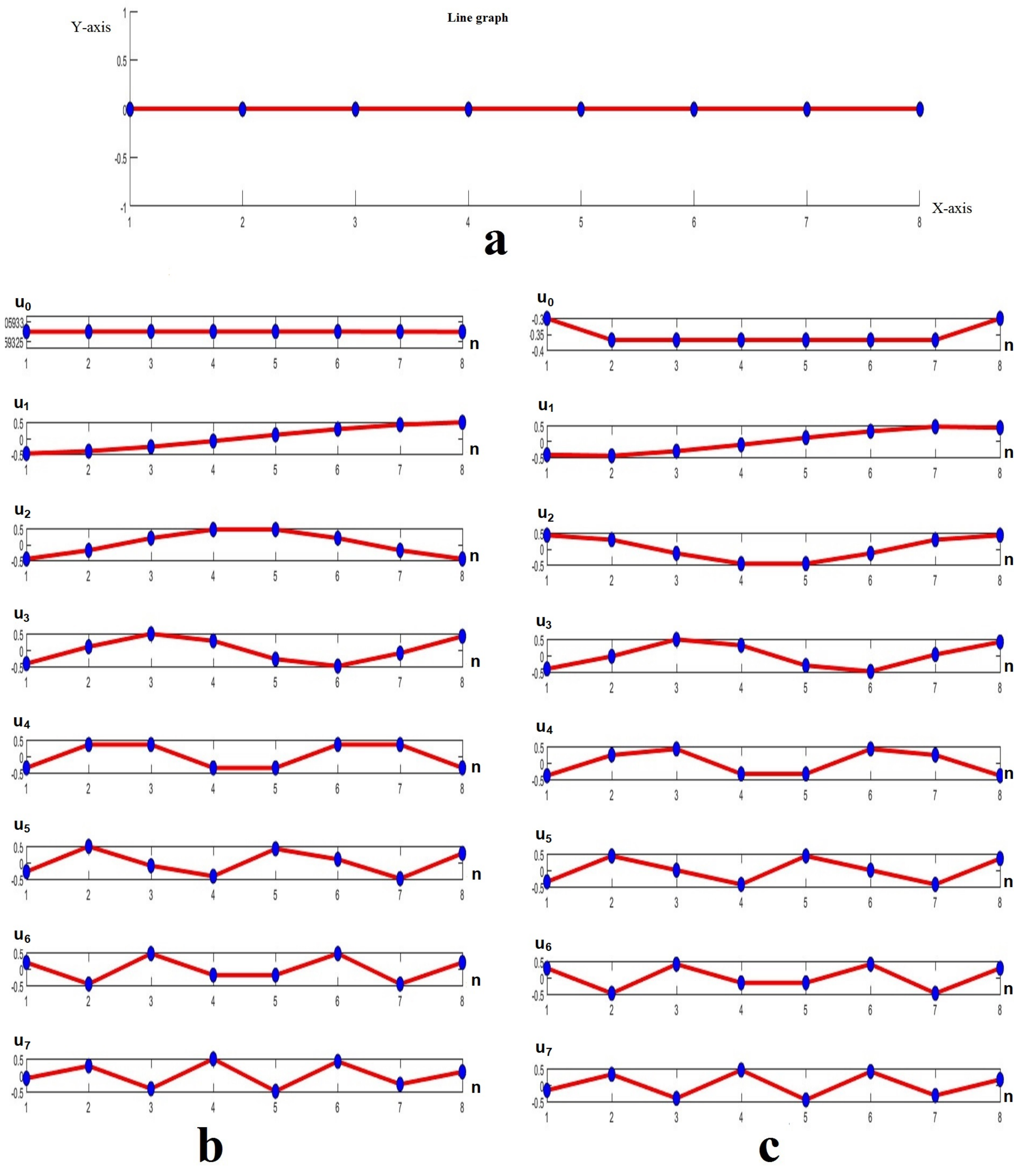


Figure 2.4: (a): Eight connected nodes as a line to form a line graph, (b): graph eigenvectors of the $\mathbf{L}$, (c): graph eigenvectors of the $\mathcal{L}$, where the X-axes refer to the individual node and the Y-axes represent corresponding eigenvector's value. This experiment shows the behaviour of the graph eigenvectors of the combinatorial (Left side) and the normalised (right side) versions for the line graph.

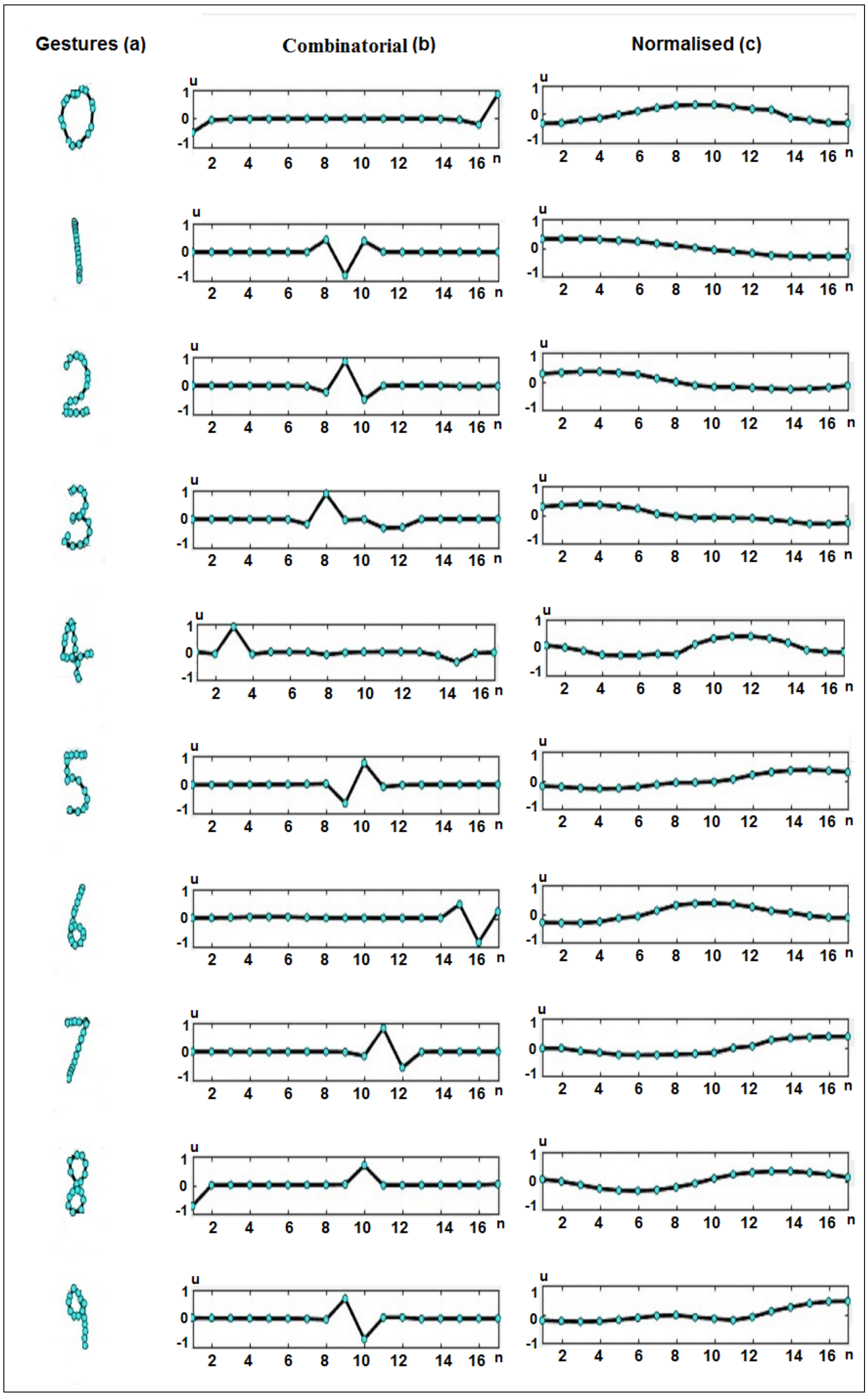


Figure 2.5: (a): Handwritten number digits, (b): the second eigenvector of their combinatorial Laplacian matrix, (c): the second eigenvector of their normalised Laplacian matrix. The X-axes refer to the individual node and the Y-axes represent corresponding eigenvector's value.

minimum value in the vector.

$$\mathbf{U}^t\mathbf{L}\mathbf{U} = \sum_{i,j=0}^{n-1} \mathbf{L}_{i,j}\mathbf{u}_i\mathbf{u}_j, \tag{2.14a}$$

$$= \sum_{i,j=0}^{n-1} (\mathbf{D} - \mathbf{A})\mathbf{u}_i\mathbf{u}_j, \tag{2.14b}$$

$$= \sum_{i,j=0}^{n-1} \mathbf{D}_{i,j}\mathbf{u}^2 - \sum_{i,j\in\mathcal{E}} 2\mathbf{u}_i\mathbf{u}_j, \tag{2.14c}$$

$$= \sum_{i,j\in\mathcal{E}} \mathbf{u}_i^2 + \mathbf{u}_j^2 - 2\mathbf{u}_i\mathbf{u}_j, \tag{2.14d}$$

$$= \sum_{i,j\in\mathcal{E}} (\mathbf{u}_i - \mathbf{u}_j)^2, \tag{2.14e}$$

Thus,

$$\lambda_1 = min \left[\frac{\sum\limits_{i,j\in\mathcal{E}} (\mathbf{u}_i - \mathbf{u}_j)^2}{\sum\limits_i \mathbf{u}_i^2} \right]. \tag{2.15}$$

Note that, ($\mathbf{D}_{i,j}\mathbf{u}^2$) is equal to ($\mathbf{u}_i^2 + \mathbf{u}_j^2$) because node i has degree D_i. So, value ${\mathbf{u}_i}^2$ needs to be summed up $\mathbf{D}_i$ times. But each $\mathcal{E}_{i,j}$ has two endpoints, so we need $\mathbf{u}_i^2 + \mathbf{u}_j^2$. Also, we know that $\sum\limits_{i=0} \mathbf{u}_i^2 = 1$ and $\sum\limits_{i=0} \mathbf{u}_i = 0$.

An example of how to use Fiedler vector for graph partitioning can be seen in Figure. 2.6.

2.2.3 Graph signal processing operations

In this section, we illustrate the basic classical signal processing applications in the graph spectral domain such as graph filtering [34], graph Fourier transforms [46], graph down sampling [37], and graph wavelets [40].

1. Graph filtering

   Image filtering can be implemented based on spectral graph theory Eq. (2.16) using a certain filter. For example, a (512×512) image is filtered using a node connection and low pass filter Eq. (2.17),

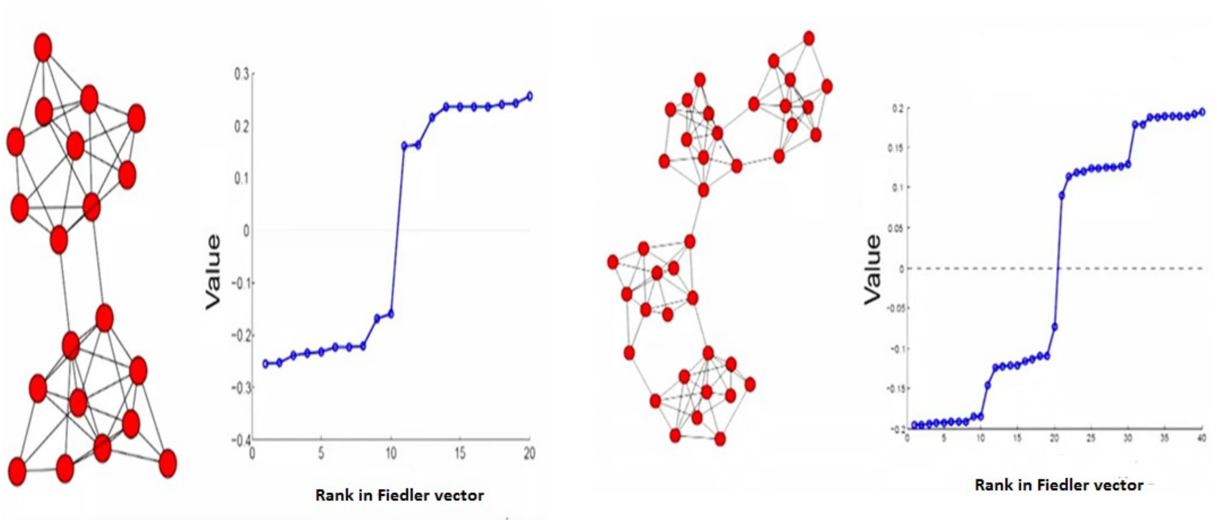


Figure 2.6: Graph partitioning using Fiedler vector [1]. In this example, two different structures and their Fiedler values are displayed, which can be used for partitioning based on the sign of the individual value.

$$\hat{f_{out}}(\lambda_\ell) = \hat{f_{in}}(\lambda_\ell)\hat{h}(\lambda_\ell), \tag{2.16}$$

$$\hat{h}(\lambda) = \mathbf{U}^t\left(1/(1+\gamma\lambda)\right)\mathbf{U}, \tag{2.17}$$

where $\hat{f_{in}}$ is the input data, $\hat{h}$ is the filter and the value of $\gamma = 10$. As shown in Figure. 2.7, we took the 80×80 pixels sub image from the cameraman image to be $\hat{f_{in}}$ and corrupted it with additive Gaussian noise with a mean of zero and a standard deviation of 0.1 to get a noisy signal. A spectral graph filtering method is applied to de-noise the signal. In this experiment, a semilocal graph is formed from the pixels by connecting each pixel to its horizontal, vertical, and diagonal neighbours (*i.e.*, eight connections). The edge weight represents the differences between neighbouring pixel values in the noisy image. Low-pass graph filtering is applied using $\gamma = 10$ to reconstruct the image. We can see in Figure. 2.7 that the graph spectral filtering method does not smooth the edges, as the geometric structure of the image is encoded in the graph Laplacian via the noisy image.

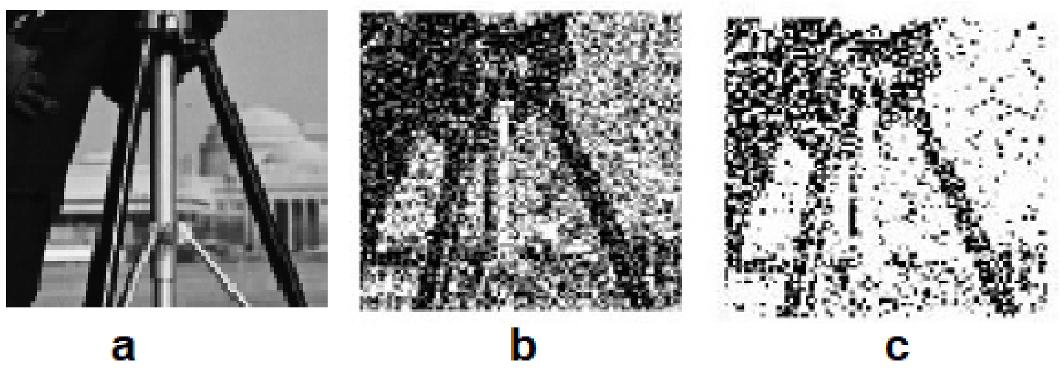


Figure 2.7: Graph filtering: (a) the original sub-image, (b) adding noise, (c) filtered image using graph spectral domain.

2. Graph Fourier transform

   Forward graph Fourier transform is calculated as shown in Eq. (2.18) [35],

$$\hat{f}(\mathcal{L}_\ell) = \sum_{i=1}^{N} f(i) u_\ell^*(i), \tag{2.18}$$

   where $*$ is a transpose operation and f is the input data. The inverse graph Fourier transform is given by:

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\mathcal{L}_\ell) u_\ell(i), \tag{2.19}$$

3. Graph down sampling

   Graph down sampling was implemented in three different ways based on; the signs of the singular value decomposition (SVD) bases [47], the colour distribution [48], and the order of the sorted nodes using maximum spanning trees (MST) [37]. MST based down sampling provides faster implementation than the SVD and colouring formulas.

4. Graph wavelet

   Several studies have proposed graph wavelet methods, and there are few differences among these, mainly being in the downsampling, compression, and multi-resolution. The basic operation of the graph spectral wavelet is to construct bases that are localised in both the graph and vertices. For example, a diffusion wavelet was performed based on the compressed power of a diffusion operation [49]. Another graph wavelet was designed

Table 2.1: Comparisons between different wavelet algorithm

	[52]	[49]	[40]	[50]	[51]
Spectral localisation	✗	✓	✓	✓	✓
Multi-resolution	Partial	✓	Partial	Partial	Partial
Down sampling	Partial	✓	✗	✓	✗
Fast transform	✓	Partial	Partial	✓	Partial
Graph compression	✗	✓	✗	✓	✗

by a two-channel filter bank as a bipartite graph [50], which demonstrated a perfectly reconstructed, orthogonal transform. However, this graph wavelet was designed for a particular case, which assumed that the graph had to be bipartite, taking advantage of the limited range of graph eigenvalues for bipartite graphs.

A spectral graph wavelet transform (SGWT) was implemented by scaling function on each vertex [40]. To simplify, its two steps were dilated and translated from a band pass kernel design in a graph for the combinatorial graph Laplacian matrix. While this method did not require any down sampling, it had a faster transform and spectral localisation. The SGWT, however, was not perfectly reconstructed, and so could only be used for a specific application that did not require this limitation. An extension of the SGWT has been proposed, where a tight frame is formed to conserve energy in the wavelet domain [51,52]. The resulting wavelet coefficients focused on developing localised transforms specifically for the data defined on the graphs. Table 2.1 illustrates the main differences between the existing graph wavelet methods.

In the following subsection, we will review the current work of the in-air hand drawing shape recognition, 2D/3D shape recognition, and hand gesture recognition. The most relevant techniques, which are used in these applications, are highlighted in Table 2.2.

2.3 In-air hand-drawn shape recognition methods

The existing work of in-air hand-drawn shape recognition is presented in this section. Common key parameters for evaluating the performance of hand-drawn methods include runtime, preprocessing, rotation invariance, and matching of different sizes. Based on these criteria, Table 2.3

Table 2.2: Several methods to classify different shapes.

	In-air shapes	2D/3D shapes	Hand gesture
Graph-based	✓	✓	✓
Model-based	✗	✓	✗
Feature-based	✓	✓	✓
View-based	✗	✓	✗
Deep learning-based	✗	✓	✓

Table 2.3: Summary of the state-of-the-art methods for in-air hand-drawn recognition.

	[23]	[25]	[30]	[31]	[53]	[54]	[55]	[56]	[57]	[58]	[59]	[60]
Different size matching			✓	✓			✓	✓	✓	✓	✓	✓
Rotation invariant	✓	✓										
Normalisation				✓	✓	✓	✓			✓	✓	✓
Downsampling			✓	✓	✓	✓		✓	✓	✓	✓	✓
Shape coding					✓							✓

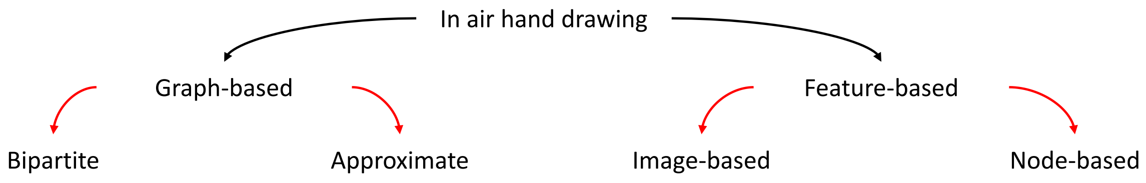


Figure 2.8: In air hand-drawn shape recognition techniques.

shows a comparison of the in-air hand-drawn shape recognition methods. These methods can be divided into two types: graph-based [31, 32] and feature-based [53, 54] as shown in Figure. 2.8.

2.3.1 Graph-based methods

Previous methods on graph matching have explored either bipartite graph matching using the vertex domain or an approximate method based on the graph adjacency matrix. In bipartite graph matching, a set of edges are chosen, with no two edges sharing the same end point, in order to maximise the matching ratio between two sets of points without increasing the degrees of the nodes. Different matching conditions have been used in bipartite graph matching such as, the shortest edge [31](Figure. 2.9), the convex path inside text images [30] and the largest

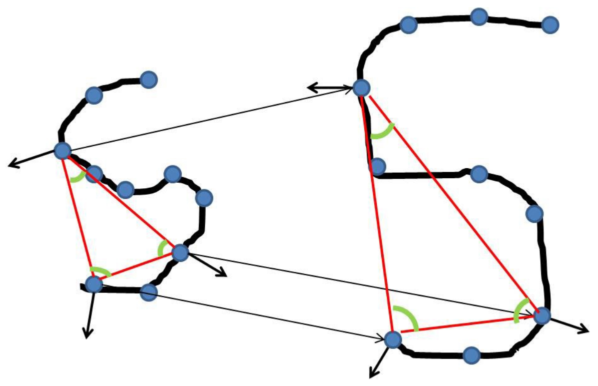


Figure 2.9: Bipartite graph matching of two samples, where the node degrees and angles connections are used to find similarities.

eigenvalue [32]. Bipartite graph matching provides real-time performance, but is sensitive to any minor changes in rotation angle.

The maximum probability of correspondence mapping between two patterns through a weight matrix is also used for matching purposes. For example, a graph indexing process is performed based both on a polynomial characterisation and a weighted graph [23, 25]. The polynomial characterisation is used in the context of polyhedral object recognition. The main limitations of these approaches are the high computational complexity and restrictions on certain graph sizes.

2.3.2 Feature-based methods

Methods of recognising hand-drawn shapes in the air can be classified into two groups: image-based representation [53–55] and node representation [56–60]. In image-based representation, the numbers are saved as an image with two types of pixels: number path pixels and background pixels. Thus, a large amount of data is required to represent the numbers, which is identified as a limitation of this approach. In one approach [55], number images are normalised into a binary table where 1 and 0 referring to the hand writing path and the background respectively. In order to eliminate the effect of too many zeros in the table, order code with shape code have been utilised [53]. Later, further improvements have been achieved by normalising the hand-path by picking out a specific number of points [54].

In node representation, the path of the hand movement is saved as a set of coordinates,

Table 2.4: The main characteristic of the existing methods of the 2D/3D shape recognition.

2D/3D shape recognition									
Deep learning		Model-based		View-based		Feature-based		Graph-based	
2DCNN [14,61]	3DCNN [62–65]	Skeleton-based [18,66–68]	Silhouette-based [69–72]	Internal 2D view [17,73]	3D view [73,74]	Global features [75–77]	Local features [16,78,79]	Bipartite [80,81,81]	Approximate [26,27,82]

Table 2.5: The main characteristic of the existing methods of the 2D/3D shape recognition.

	Deep learning	Model-based	View-based	Feature-based	Graph-based
Global detection	✓	✓	✓	✓	✓
Local detection	✓			✓	
Rotation invariant	✓				✓
Real-time performance				✓	
Different size matching	✓	✓	✓	✓	
Small amount of required data		✓	✓	✓	✓

which are captured by different ways, such as, tracking the hand node using Kinect skeleton representation [56–58], tracking the position and the orientation of the user's hand using a Wii remote [59], or using supervised learning to detect hand path [60]. The drawback of these methodologies is the sensitivity to any slight change in the angle of drawing, which causes a serious mismatching of the samples [55]. Also, these low-level features are extracted from pixels, requiring a large amount of data to achieve the optimal level of accuracy. This has hindered their ability for real-time operation.

2.4 2D/3D shape recognition methods

Although considerable progress has been made in shape-matching [4,83–86], an efficient recognition system has yet to be developed. This is because the number of complex shapes is unlimited, and each has their own unique properties, making the matching process a difficult task to perform. 2D shapes are usually presented as a 2D binary or RGB image, while 3D shapes are often presented as a 3D point cloud. To identify such shapes, many concepts have to be considered in the matching method, such as global details, local details, rotation invariance, data requirements, size, and performance time. Available methods in the literature can be divided into five categories, including: graph-based [26,27], model-based [18,66], feature-based [16,78], view-based [17,73], and deep learning-based [14] as illustrated in Table 2.4. Table 2.5 highlights the main characteristic of the existing techniques for 2D/3D shape recognition.

2.4.1 Graph-based

Graph-based approaches usually generate a graphical model based on an abstract shape or a skeleton representation to imitate the shape configuration in 2D or 3D space. Similarly, an approximation graph matching method can be applied to identify 2D/3D shapes by exploiting the adjacency matrix of the shape. Current techniques explored the maximum probability of correspondence mapping between two patterns through the weight matrix based on the eigen domain [27], spectral relaxation [26], and higher order constraints [82].

A few works (*e.g.*, [80, 81]) have explored the utility of bipartite graph matching for 3D shape recognition. For example, a set of contour descriptors and the interior region of a 3D object was extracted in order to employ a greedy bipartite graph matching algorithm [80]. In addition, a fully dynamic and fast bipartite graph matching algorithm was used for 3D shape recognition based on the polynomial deterministic [81].

The main problems associated with the graph-based approaches are the execution time of the approximation methods, making it unsuitable for real-time applications; and that local details are often ignored in bipartite graph matching, making high similarity shape matching difficult task to achieve.

2.4.2 Model-based

Model-based methods create a model representation in order to characterise the geometric details of the shape by extracting local, global, skeleton, or silhouette features. Such models involve two types of approaches: shape-skeleton studies and shape-contour studies. Skeleton-based studies mainly involve constructing a tree model using object's edges to form a shape descriptor, with the similarity measurement being based on tree matching approaches. For example, different methods can be implemented by creating a shape descriptors prototype using: short-cut [18]; points corresponding [66]; skeleton pruning [67]; and shape scaling [68]. Alternatively, several studies have relied on the boundaries provided by silhouette images. These edges efficiently characterise the global structure of an object using a single closed curve, as long as there are no holes inside the object. An early study by Zahn and Roskies [69] used Fourier descriptors to represent the shape, while the latest studies are based on progressive shape-distribution-encoding [70]; structure integral transform [71]; and chordal axis transform (CAT) [72]. In general, these approaches can provide a rich understanding of the global shape structure. However, the main issue with the model-based methods is that local details are omitted or entirely neglected. In other words, most model-based methods ignore small protrusions or dense areas, focusing only on the global structure. For each method, a specific dynamic programme was used to identify the matching similarities between patterns. In some cases, even if the algorithm was not optimised, the matching programme may be able to increase the recognition score.

2.4.3 Feature-based

Feature-based methods involve extracting a set of features that provides an optimal representation of the shape structures followed by a classification algorithm for matching process. Feature-based studies typically require more than one feature to describe a complex structure and the vast majority of literature falls into this category using a variety of features. Such features may capture the local details using scale invariant feature transform (SIFT) [78]; tree union [16]; local phase [79]; shape histogram [87]; part decomposition [33]; or the global shape characteristic such as; contour features [75]; inner-distance [76]; full shape [77]; bag of words [88]; curvature [89]; compact bag-of-features heat kernel signatures (CboFHKS) [90]; Fourier descriptors [91]; clustering coefficient [26]; and hierarchical structure [15]. These methods demonstrate a great achievement in terms of accuracy, complexity, and capturing the topology details. However, identifying shapes from different angles is the only challenging problem for these methods.

Since graph representation is invariant to rotation, using a hybrid method to extract features based on a graphical model can enhance the performance of both types of methods. Therefore, a recent work using complex-network [92] shows an efficient performance for 3D shape recognition. In this study, graphs are created to capture the global shape. This is followed by a graphical growing procedure based on pre-determined threshold in order to highlight the local details. However, since shapes have different scales and articulations, specifying one threshold value for all shapes in datasets is not an optimal way to capture the topology description.

2.4.4 View-based

View-based methods rely on similarity measurements between models from all viewing angles. In other words, the similarity score is based on how similar the view angle is between two models. View-based approaches are usually applied in 3D shape recognition [73, 74]. Only a few studies have applied visual similarity techniques in 2D shape recognition. For example, using several views generated from a circle centred on a shape's centroid, based on distance from each viewing point [17]. Another method [73] was used to generate a 3D shape prototype by viewing the model from several angles. The resulting 2D images are then interpreted to create a single 3D model. A hybrid method using view similarity and deep learning [74] has also been employed for shape recognition, in which the input data to the neural network are the shape segmentations. Similarly, multi-view depth line (MDLA) [93], symmetric branch [68], and complex function [94] can be used to obtain an understanding of topological details from different angles. However, most studies have focused only on matching the global shape, resulting in a mismatch between shapes with similar outlines. Also, typical approaches have been performed using analysis of multiple views of a shape, which leads to high complexity.

2.4.5 Deep-learning-based

The robustness and dominance of deep-learning approaches in shape recognition have been proven in the literature. Since this thesis explores the graph spectral feature representation, deep learning methods are considered beyond the scope of this thesis. Therefore, the common approaches to 2D/3D deep learning shape recognition methods are referred to only briefly in this section. For example, a deep auto encoder four-layer coding network is implemented to retrieve shapes [14] using deep learning. 2D face and shape images are classified based on a series of images, extracted from different angles [61, 74]. Similarly, a 3D filter and pooling operations are implemented in 3D point cloud [64, 65]. A volumetric representation of a shape obtained through a convolution process has been demonstrated to be a powerful tool in deep-learning approaches because it contains the local and global details of the shape [62,63]. In other words, the filter design is important in capturing structural details for deep-learning approaches.

Although such methods have a high level of accuracy in both 2D and 3D shape recogni-

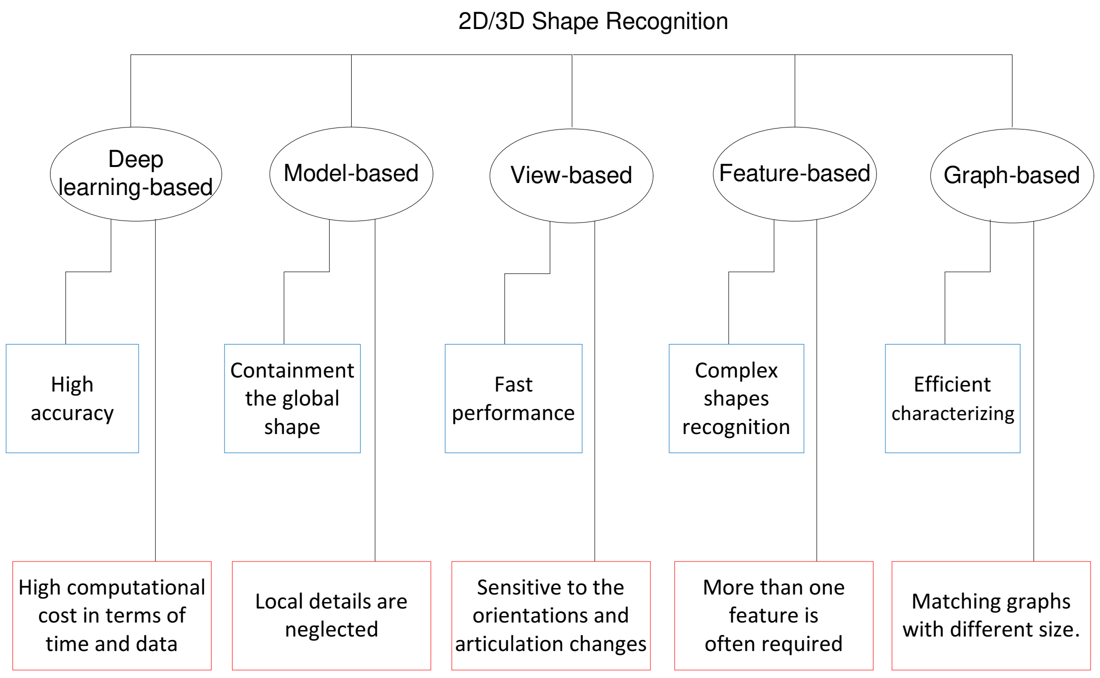

2D/3D Shape Recognition
Deep learning-based
Model-based
View-based
Feature-based
Graph-based
High accuracy
Containment the global shape
Fast performance
Complex shapes recognition
Efficient characterizing
High computational cost in terms of time and data
Local details are neglected
Sensitive to the orientations and articulation changes
More than one feature is often required
Matching graphs with different size.

Figure 2.10: The current work of the 2D/3D shape recognition and their main advantages (blue boxes) and disadvantage (red boxes).

tion, they are still not suitable for all applications due to the amount of training time and data required. In addition, there is no strong evidence available to help in determining the most efficient parameters for designing the networks. The performance of the existing works rely entirely on the experimental setup and authors skills. The main advantages and disadvantages of the 2D/3D techniques can be summarised in Figure. 2.10.

2.5 Static hand gesture recognition

Many algorithms have been proposed in recent years to address the optimal hand gesture recognition method and a comprehensive review can be found in [5,95,96]. In general, a hand gesture recognition system begins by capturing the hand data, then uses pre-processing steps to extract the hand's descriptors, which are classified as shown in Figure. 2.11.

Previous studies on hand detection have relied on colour data (*e.g.*, skin colour) to segment the hand. However, colour-based hand segmentation is not an optimal solution because different hand's have different skin colours, and there can also be similarities in face and hand

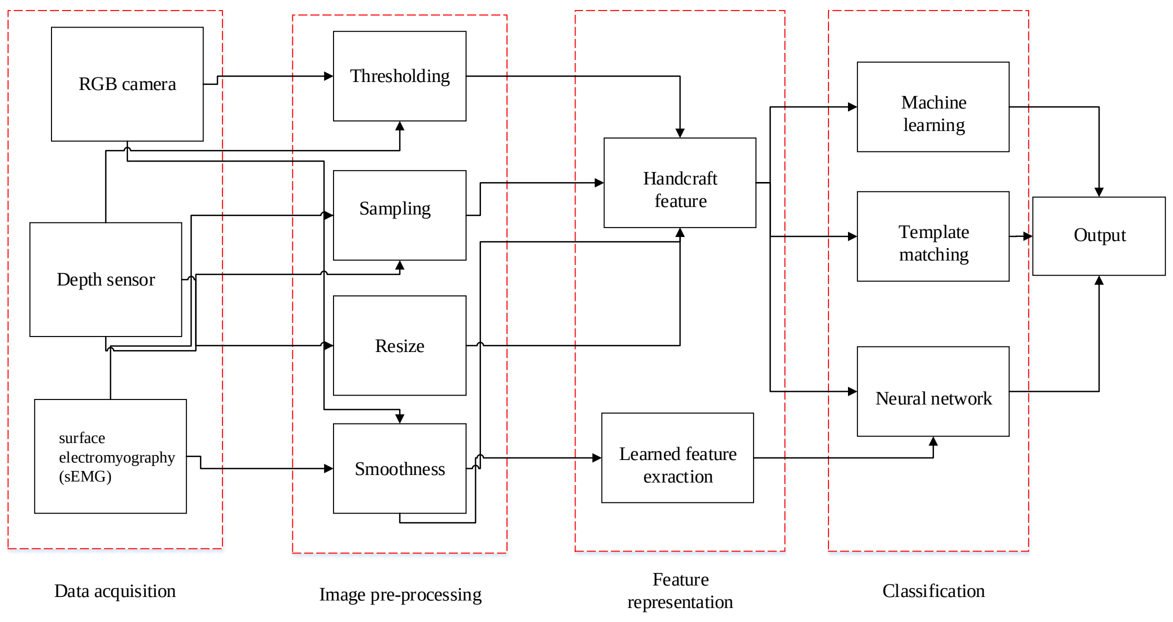


Figure 2.11: A standardised pipeline for a manual hand gesture recognition studies, which can be categorised primarily into four steps (data acquisition, pre-processing, feature extraction, classification).

colour with the background. The recent development of cameras (see Figure. 2.12), such as the Kinect sensor, Time-Of-Flight, and Real Sense have contributed to the establishment of an appropriate communication between humans and machines. In particular, depth information can reduce many of these issues related to colour-based hand detection such as lighting conditions and background clutter. Depth maps provide an efficient method for detecting and segmenting the hand. Several methods have been proposed for acquiring accurate depth information. For example, three different views of the hand were analysed to obtain an accurate model of a hand [97]. Others have assumed that the hand is the closest object to the camera in a scene [98]. A combination of colour and depth mapping can also be used for hand detection [99]. Although combination colour and depth images have resulted in great improvements in hand detection, the time requirements for processing both the cameras and the calibration process in order to align both cameras are a main concern in this technique.

Once the hand is detected, preprocessing steps are performed to eliminate noise (*i.e.*, using thresholds) and obtain a consistent representation of the hand (*i.e.*, using a sampling and resizing procedure). The preprocessing steps are used in almost all current studies. The hand

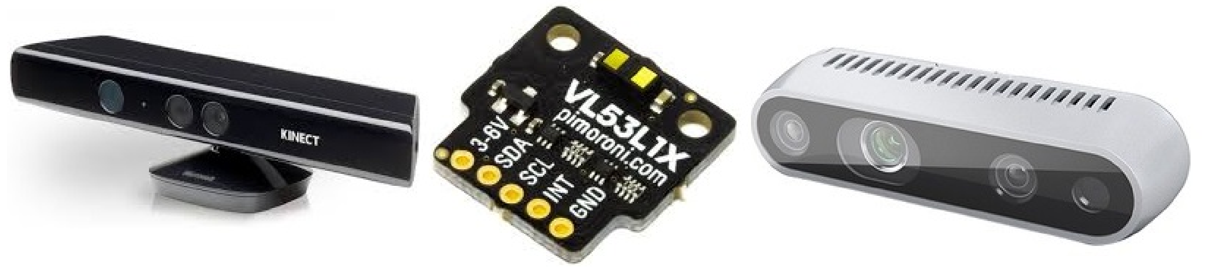


Figure 2.12: Depth sensors from left to right: Kinect sensor, time of flight (ToF) sensor, Real Sense.

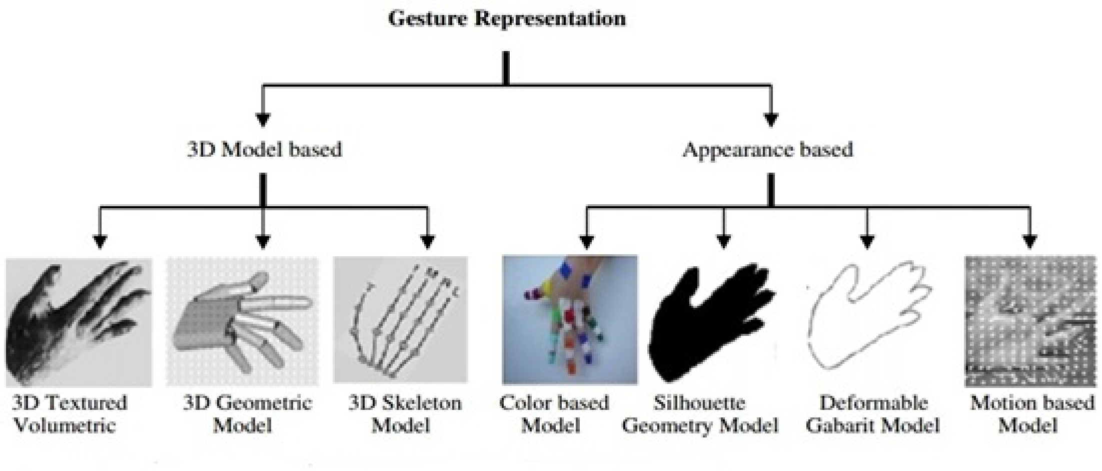


Figure 2.13: Different ways of hand representation [2].

representation resulting from this step can be broadly categorised into two types, appearance-based and 3D model-based as shown in Figure. 2.13. Producing a 3D hand model involves three techniques: a 3D texture volumetric, 3D geometric, and 3D skeleton models. Such representation is usually used for virtual hand generation [100, 101], while hand gesture recognition studies utilise appearance-based representation. A 3D model-based usually shows the 3D spatial description of the hand along with temporal aspects. Such model provides details about each finger on the hand individually or in combination with the other fingers. The advantage of this type of representation is the ability to update the hand parameters during tracking by the camera, leading to precise hand gesture representation. The 3D texture volumetric model contains more details about a hand skin and skeleton.

Hand representation is followed by a feature extraction process, which is the most important step. Hand gesture features can be divided into two types, learned and handcraft features.

Learning feature aims to obtain features, extracted from initial trained hand data and use it to identify completely new data for the model [102]. Hand craft features are a set of descriptors that represent the shape. They can be categorised into two types: graph-based features, and general feature-based methods.

2.5.1 Graph-based methods

Previous works using graph-based methods have been applied only in static hand gesture recognition [103–108] employing graph theory rather than the spectral domain. For example, a Gabor filter is applied over the hand in order to allocate the regions of interest [103–106]. By considering these regions as nodes, a graph is created to capture the hand topology. The similarity measurement is based on the degree of nodes and the graph size. The main issue with these studies is in the hand detection. Since these are colour-based methods, mixed detection can occur in real-life applications. For example, the face and any similar colour in the background might be detected as one object. Another graph-based hand gesture recognition method [107] performed a tree representation over the hand. In this study, the fingers were connected in a pairwise Markov random field, which enforced the connectivity of the hand structure through soft constraints. Pairwise finger connections overcome the occlusion issue in hand gesture recognition. More recently, a combination of hand appearance and graph features (GA) [108] has been utilised for RGB hand gesture recognition. A growing neural gas network was used to imitate the topology of the hand, and the resulting nodes were used for graph generation. Then, different classes were identified based on statistical measurements between the nodes, such as the number of nodes, and the angle of node distribution. Such studies, however, would have been more robust if they had applied a spectral-domain because this would have provided an efficient characterisation of the structures.

2.5.2 Feature-based methods

The majority of work on hand gesture recognition rely on extracting a set of descriptors to enable distinction between different classes. These descriptors are based on either skeleton or silhouette representations. The advantages of a skeleton representation includes providing

Table 2.6: Common classifiers in hand gesture recognition methods.

	prediction speed	Memory capacity	preferred for
Support vector machine (SVM)	Slow	Medium	Two classes, linear data
Neural Network (NN)	Slow	Medium	General
Random Forest (RF)	Fast	Small	Non linear data
K-Nearest Neighbour algorithm (KNN)	Medium	Medium	Euclidean distance
Dynamic time warping (DTW)	Fast	Medium	Two different length signal
Hidden Markov Model (HMM)	Medium	Large	Previous case estimation

accurate tracking of the fingers and no assumption that the hand must face the camera. Therefore, the literature on hand-skeleton methods includes a variety of features such as, histogram of 3D facets (H3DF) [109], histogram of oriented gradients (HOG) [110], dimensional zernike moments (DZM) [111], zernike moments (ZM) [112], Fourier descriptors (FD) [113], local curvature with distance transform (LCDT) [102], local energy function [114], and depth project map based bag of contour fragments (DPM-BCF) [115]. Two problems associated with such studies include, the complexity of finger detection and locations. This is because finding only the number of extended fingers is not enough to categorise the shape, unless there is information about the location of each extended finger.

The silhouette feature has also been considered in a number of studies and shows improved performance using the finger earth mover's distance (FEMD) [99], super-pixel earth mover's distance (SP-EMD) [116], and finger segmentation (FS) [117]. However, silhouette techniques are affected by the hand orientations, being required to capture the correct gestures and the current work has only been tested against minor changes in rotation angle [99, 116]. The low resolution of the depth stream is another issue, making an accurate finger detection a challenging task. Moreover, most of the works mentioned above use depth information, involving intensity of the Gray image for hand segmentation only.

In the final step, these features are classified using either template matching (TM) [99], dynamic time warping (DTW) [98], nearest neighbour (KNN) [110], support vector machine (SVM) [111], linear discriminant analysis (LDA) [108], or hidden Markov model (HMM) [113] methods. A brief comparison between the common classifiers for hand gesture recognition can be found in Table 2.6.

2.5.3 Deep learning-based methods

Only a few studies have used deep learning for hand gesture. This is mainly because of the time and data requirements [118–122]. Moreover, feature-based methods have already achieved a high level of accuracy with real-time performance, and this has reduced the need for a deep-learning approaches. Inputs to the neural network includes two types of data either images [118–120] or surface electromyography (sEMG) signals [121, 122], that obtained from the user's hands. The sEMG-based method has demonstrated optimal performance despite the state of the surrounding environment (*i.e.*, lighting conditions, camera scene, and noise). However, such methods are usually evaluated using a small numbers of classes, and the performance decreases when the categories are increase.

2.6 Concluding remarks

In this section, we have outlined the categorisation of shapes into three different types based on their structural complexity. We have also presented basic information about graph theory, graph spectral theory, relevant works on in-air hand drawn shape recognition, 2D/3D shape recognition, and hand gesture recognition. Based on the existing works, we draw several conclusions:

1. The existing graph-based approaches suffer from time complexity and rotational changes. To solve these issues, in **Chapter 3**, we explain that performing feature-to-feature rather than point-to-point matching reduces the time complexity. Moreover, since the representation of edges depends on relative measurements between the nodes, they are not affected by scaling or angles changes.

2. Covering intra-class variations in high similarity shapes is a challenging task in 2D shape classification. Therefore in **Chapter 4**, we introduce a new method for generating adaptive graph connectivity parameters to capture small articulations.

3. With more complex shapes, it is necessary to obtain a deeper understanding of the details. This can be achieved by simplifying the shape, along with analysing the global and local details of the structure. Thus, in **Chapter 5** a new method is presented for 2D/3D

shape recognition that involves capturing the local and global details of the structure and simplifying objects using a graph partitioning technique.

In the next chapter, we will introduce the in-air hand-drawn shape recognition work.

Chapter 3

Graph spectral domain based on global details for in-air hand-drawn recognition

3.1 Introduction

As mentioned previously, this chapter proposes a novel method for in-air hand-drawn shape recognition. Human computer interaction has become popular in applications, such as, interactive computer gaming, robotics, sign-language recognition, healthcare and assisted living mainly due to the availability of inexpensive depth sensors, such as, Kinect sensor. Recent years therefore have seen new advances in hand movement understating methodologies [99, 116, 123, 124]. These studies were primarily focussed on either static hand gestures or dynamic hand gestures, such as hand movements. Many modern applications rely on direct observation and analysis of the human body part movement. The smooth flow and flexibility of the human body configuration allow the humans to create very complex and arbitrary shapes in the air using hands. Automated analysis and understanding of these complex shapes created by hand movements benefit in real-time gesture-based human computer interaction applications.

Particularly, in-air hand-drawn number recognition has attracted a strong attention [53–60] due to applications-related importance and the interesting challenges in the problem. These methods are based on the image representation of the numbers [53–55] followed by shape matching or the path representation of the hand movement [56–60]. However, all these methods suffer from the sensitivity to changes in the angle of drawing leading to serious recognition errors of the in-air drawn.

To solve these issues, in this chapter, we exploit the emerging graph signal processing (GSP) to propose a graph spectral feature representation for in-air drawn arbitrary shape recognition. The GSP has attracted a great attention in processing, analysis, coding and understanding of data sampled on a non-uniform grid, often referred to as irregular data or graph data. The classical discrete signal processing is not directly applicable on such irregular data. GSP provides a robust mechanism to represent irregular data in terms of their connectivity to each other when represented as a graph. The connectivity among the vertices characterises the global structure of the graph and it does not change after rotation, flipping or mirroring of the graph structure. Since graph spectral representation is driven by the connectivity, rotation invariant features can be extracted in the graph spectral domain. Therefore, the proposed method, that explores such features, is not sensitive to the drawing angle. In terms of sampling the hand movement paths of the in-air drawn shapes, we aim to minimise number of vertices while keeping the properties of the structure intact. This leads to lowering the complexity without affecting the recognition accuracy rates. The use of the graph matching on vertex domain has been explored for shape matching purposes in the literature [30, 31, 125]. However, they are not robust to variations in angles of orientation of the shapes and numbers.

Thus, this chapter presents a novel set of the graph spectral domain features representation for accurate and fast in-air hand-drawn number and shape recognition. In this work, we explore representing the shapes and numbers in the graph spectral domain as opposed to the node domain for feature extraction for recognition. Different shapes are represented based on the connectivity description through the graph spectral domain. A fixed number of features is extracted followed by machine learning for recognition of shapes. The proposed method is experimented using Kinect sensor for data capturing and real time recognition (as in Figure. 3.1). A demonstration video of this method can be found at [126]. The main contributions of this work are:

- Proposal of a new set of graph spectral domain features for in-air hand-drawn number and other shape recognition.
- Proposal of a new rotation and flip-invariant feature set with real-time operation.

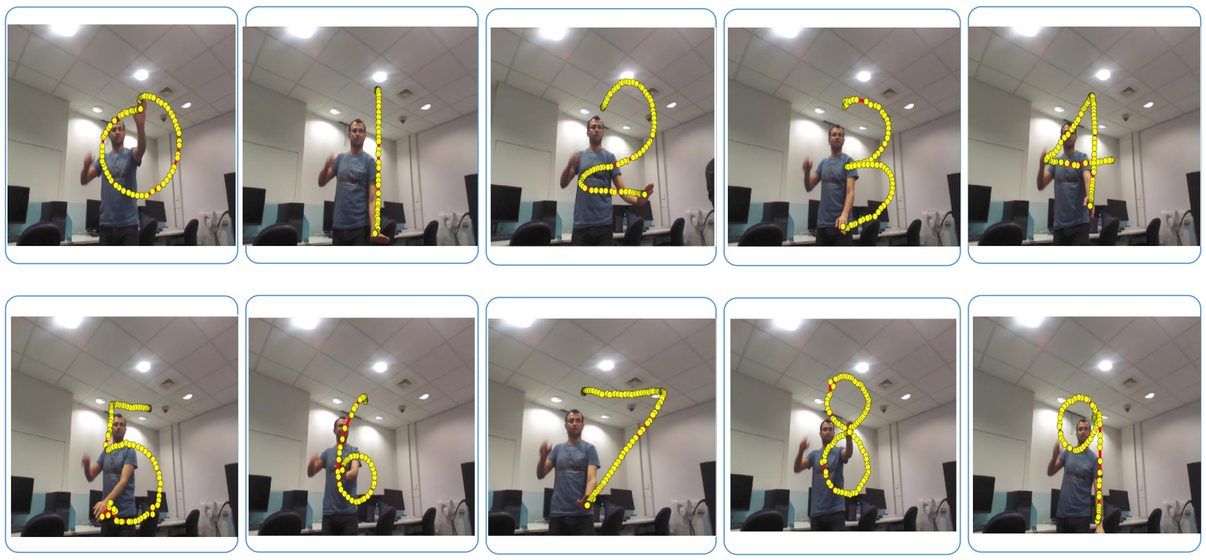


Figure 3.1: Left–right flipped screen shots of in-air drawing.

- Creation of a new dataset for in-air hand-drawn number and shape recognition research [127].

The rest of this chapter is organised as follows: Section 3.2 presents the proposed graph spectral features in the context of a in-air hand-drawn number recognition system. Section 3.3 shows the results and discussions in terms of number and other shape recognition followed by the concluding remarks in Section 3.4.

3.2 The proposed method

The proposed method (Figure. 3.2) can be divided into four steps: data acquisition; pre-processing; graph spectral feature extraction, and classification. The main novelty in the work presented in this chapter is in graph spectral feature extraction. Details of data acquisition and pre-processing are included in this section for completeness. However, it should be noted that the graph spectral feature extraction and classification methodology can be applied on any node representation of in-air drawn numbers and arbitrary shapes.

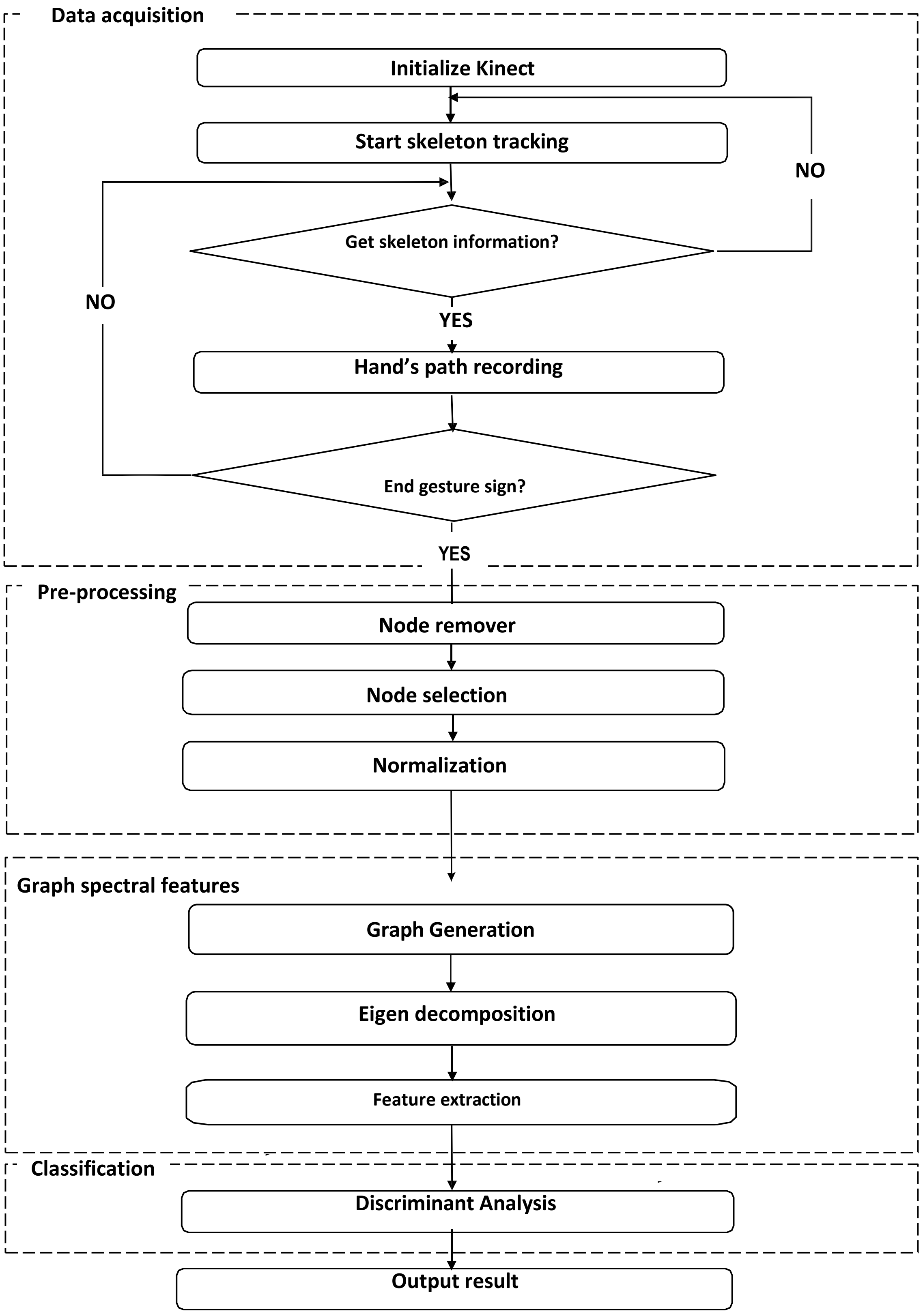


Figure 3.2: Flowchart of the proposed method.

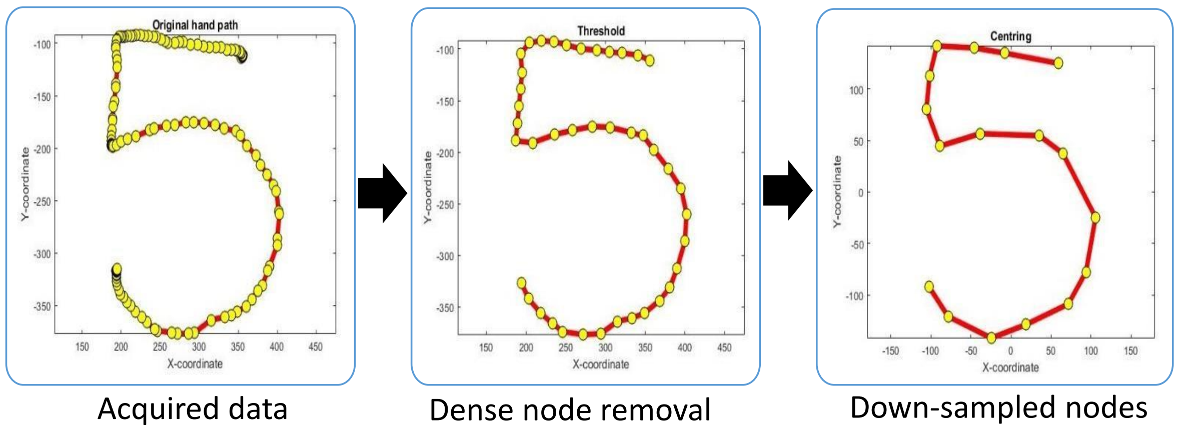


Figure 3.3: Data acquisition and pre-processing steps: (Left) Data acquired from the original hand path; (Middle) Dense nodes removal; (Right) Node Down-sampling.

3.2.1 Data acquisition and pre-processing

The Kinect sensor is used to acquire depth data based on skeleton tracking. Users have to stand in front of Kinect around 1 to 3 metres away [128]. The right hand is used to draw numbers in-air. Then, users can raise their left hand higher than the shoulder to end the movement as in Figure. 3.1. By doing so, users will have unlimited time to draw digits in-air. The depth values acquired using the Kinect skeleton tracking is not accurate when the hand movement is fast. Therefore, several methods have been proposed in the literature to get more accurate depth information, such as, 3D analysis [97], combining of colour and depth information [99] and considering the closest object to the camera in the scene [98]. In this work, hand contour searching is performed based on the assumption that the hand is the closest object to the camera in the scene. The search takes place in a block of 20×20 pixels. In this case, the block centre represents the right hand position. The left hand tracking is based on the Kinect skeleton tracking because this technique is fast and no depth information is used for the left hand operation.

The acquired data can be densely sampled as can be seen in the left sub-figure in Figure. 3.3. For each point, the (X, Y) coordinates on the vertical plane and the depth Z are recorded to form the 3D measurement space in (X, Y, Z). In order to remove these unwanted samples, starting from the first sample, for each sample, all the samples captured within a distance less than a chosen fixed threshold (T) are removed. This process eliminates the extra nodes that are created when the user stops moving their hand at any point during drawing or at the end of drawing. The resulting in-air drawn path, P, is a smooth curve with N number of nodes as can be seen in the middle sub figure of Figure. 3.3. To reduce the complexity of the subsequent graph spectral decompositions, we choose n number of nodes, where $n < N$, to form a new down-sampled path, $\hat{P}$, as follows (as in the right sub figure of Figure. 3.3):

$$\hat{P}(i) = P\left(\left\{\frac{iT}{n}\right\}\right), \tag{3.1}$$

where $i = 0, 1, ..., n-1$ is the new node index and $\{\}$ is the rounding to the nearest integer. This is followed by normalising which includes centring the down-sampled path, $\hat{P}$, around the centre point (0,0,0) and resizing the (x, y) coordinate range to $m \times m$. Figure. 3.4 shows an example of the normalisation process. Finally, the node i represented with its coordinates (x_i, y_i, z_i).

3.2.2 Proposed graph spectral features

We can now represent the nodes in the hand path, $\hat{P}$, as the nodes in an undirected and fully connected graph, $\mathcal{G}$. The weight, $\mathbf{A}_{i,j}$ corresponding to an edge, $e_{i,j}$ is computed as in Eq. (2.1).

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

$$\hat{\mathbf{r}}_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \qquad i = 0, 1, ..., n-1. \tag{3.2}$$

An example of a $\mathcal{G}$ and its signal $\hat{\mathbf{r}}$ is shown in Figure. 3.5. We also define the signal $\theta : \mathcal{V} \rightarrow \mathbb{R}$, where i^{th} component represents the angle of the vertex i in $\mathcal{V}$ with respect to the centre (0,0,0) as follows:

$$\theta_i = tan^{-1}\left(\frac{|y_i|}{|x_i|}\right). \tag{3.3}$$

The absolute value of x_i and y_i keeps the range of the angle between $(0° - 90°)$. For example, the points (3,4), (-3,4), (3,-4) and (-3,-4) have the same angle value, which is equal to 53.13°. An example of a $\mathcal{G}$ and its signal θ is shown in Figure. 3.5.

The combinatorial graph Laplacian matrix, $\mathbf{L}$, is defined as in Eq. (2.3) and the diagonal matrix is computed as in Eq. (2.4). As the shapes form non-regular graphs, we consider the

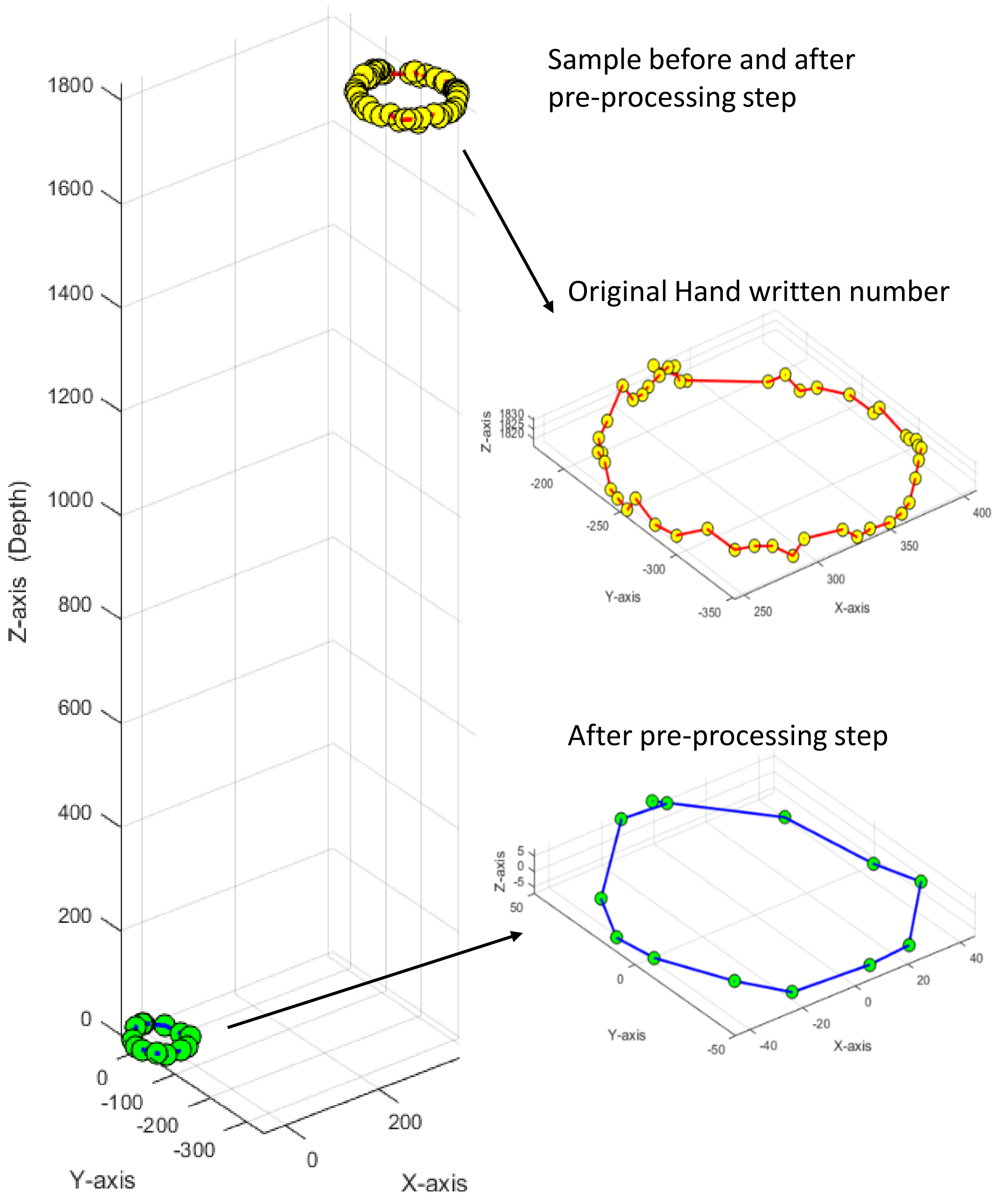


Figure 3.4: 3D space of a digit before and after pre-processing. Original in-air hand drawing sample with its zoomed version is plotted in yellow, and the result of preprocessing steps including remove noise, down sampling and centring is plotted in green.

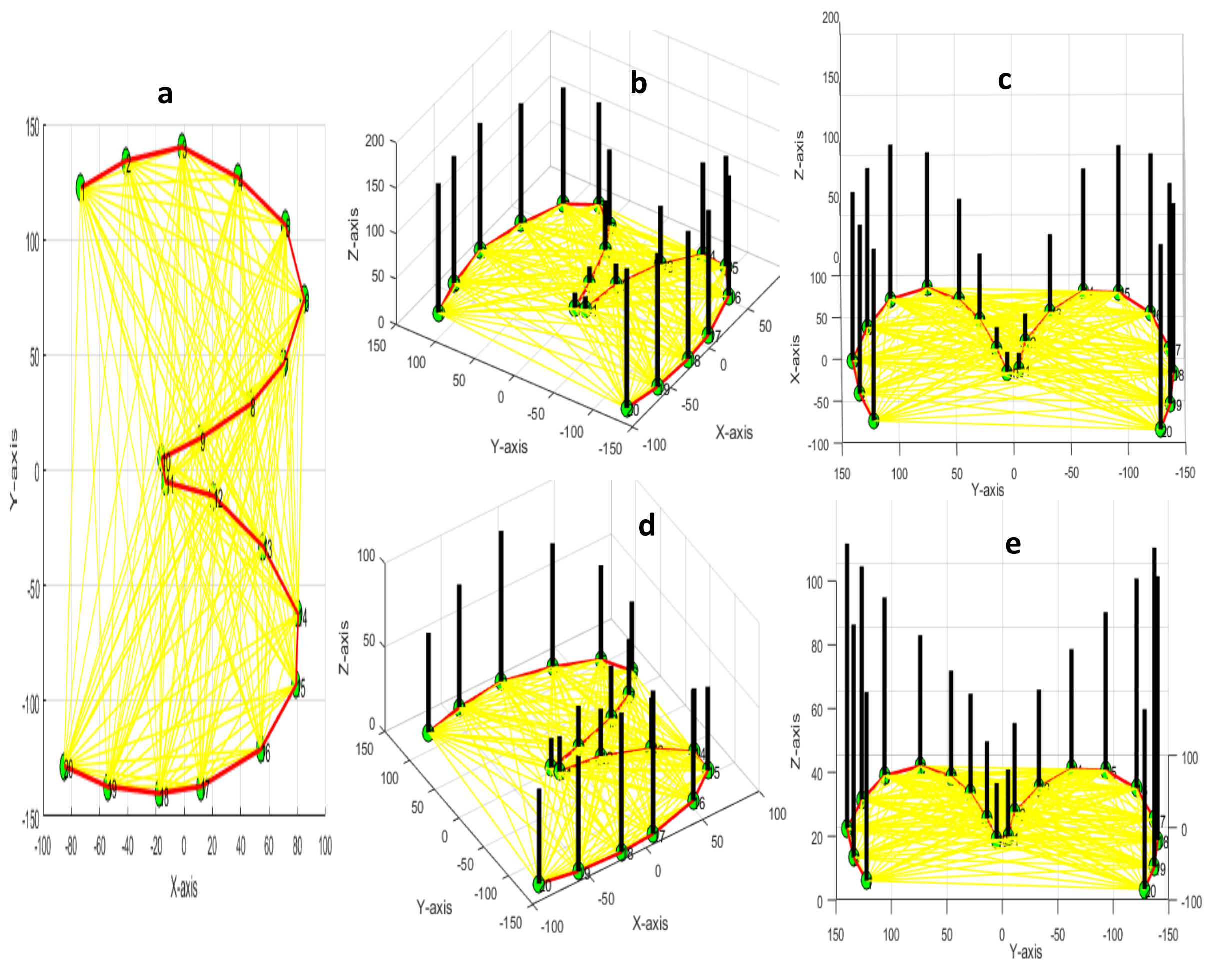


Figure 3.5: Graph construction above the hand path (red lines) with its vertices (green points), fully connected node edges, $\mathcal{E}$, (yellow lines) and the node values (black lines), where (a): Top view of the graph, (b): $\hat{\mathbf{r}}$ values (Side view), (c): $\hat{\mathbf{r}}$ values (45° view), (d): θ values (Side view), (e): θ values (45° view).

symmetric normalised Laplacian matrix, ($\mathcal{L}$), and the geometric version as shown in Eq. (2.5) and Eq. (2.6). The spectral decomposition of the graph is computed as in Eq. (2.7).

Although, $\mathbf{L}$ has been widely used in image processing related applications [40, 41], the symmetric normalised Laplacian, $\mathcal{L}$, is thought to be more appropriate for representing non-regular graphs in the literature [39, 45]. Therefore, in this work our primary focus is on the symmetric normalised Laplacian.

The graph eigenvectors have been used in analysing graph spectra both algebraic and analytic wise [129,130]. It has been shown in [129], that given a graph with no isolated vertices and $\mathcal{L}\mathbf{u} = \lambda\mathbf{u}$, *i.e.*, $\mathbf{u}$ is an eigenvector of $\mathcal{L}$, then the corresponding harmonic eigenvector, $\mathbf{y}$ associated with the eigenvalue λ is $\mathbf{D}^{-\frac{1}{2}}\mathbf{u}$. Thus, given the harmonic eigenvector for $\lambda = 0$ defined as $\frac{1}{\sqrt{n}}\mathbf{1}$, where $\mathbf{1}$ is the constant 1 vector and n is the number of nodes, the first eigenvector, $\mathbf{u}_0$, is defined as

$$\mathbf{u}_0 = \mathbf{D}^{\frac{1}{2}}\frac{1}{\sqrt{n}}\mathbf{1}. \tag{3.4}$$

The remaining eigenvectors are orthogonal to $\mathbf{u}_0$, since $\mathcal{L}$ is symmetric. As can be seen from Eq. (2.4) and Eq. (3.4), the eigenvector $\mathbf{u}_0$ of $\mathcal{L}$, corresponding to the lowest eigenvalue, $\lambda_0 = 0$ captures important details of the structure of the graph. Therefore, in our proposed method, we explore the use of $\mathbf{u}_0$ as part of features for shape recognition.

Many experiments are conducted in order to achieve effective discriminatory features of all structures. We start with one simple feature such as, the graph eigenvector matrix, individual graph eigenvectors, $\mathbf{u}_0.i$, $\mathbf{u}_0.i^2$,$\mathbf{u}_0.i^3$, $\hat{\mathbf{r}}$, and θ. We found that $\mathbf{u}_0.i^2$, $\hat{\mathbf{r}}$, and θ provide the highest level of accuracy. Therefore, we carry out additional experiments to investigate the performance of a combination of features. More details about these experiments can be found in the appendices Section A. At the end, we propose a feature vector comprising of the following 3 components:

1. The first part of the feature vector addresses translation invariance of the shape analysis of $\mathbf{u}_0$ by computing the second moment components about the mean as follows:

$$\mathbf{F}_1 = \mathbf{M}_1.\mathbf{u}_0, \tag{3.5}$$

   where modulation matrix, $\mathbf{M}_1$, is a diagonal matrix with diagonal elements computed as follows:

$$\mathbf{M}_{1_{(i,i)}} = (i + 1 - (n+1)/2)^2 + 1, \qquad i = 0, 1, ..., n-1. \tag{3.6}$$

2. The second part of the feature vector modulates $\mathbf{u}_0$ with the distance to each node with respect to the origin, (0,0,0), as follows:

$$\mathbf{F}_2 = \mathbf{M}_2.\mathbf{u}_0, \tag{3.7}$$

where the modulation matrix, $\mathbf{M}_2$, is a diagonal matrix with diagonal elements are formed by the magnitude signal, $\hat{\mathbf{f}}$, computed by Eq. (3.2).

3. The final part of the feature vector modulates $\mathbf{u}_0$ with the angle to each node with respect to the origin as follows:

$$\mathbf{F}_3 = \mathbf{M}_3 . \mathbf{u}_0, \tag{3.8}$$

where the modulation matrix, $\mathbf{M}_3$, is a diagonal matrix with diagonal elements are formed by the angle signal, θ, computed by Eq. (3.3).

Overall, the feature vector is formed by concatenating the three vectors, $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$, resulting in a feature vector with total length equal to $3n$. The components of the feature vectors for numbers 0 to 9 are shown in Figure. B.22. Similarly, Figure. B.23 shows the feature vector components for the shapes included in our dataset, which is shown in Figure. 3.6. More details about the proposed feature can be found in the appendices Section B.

3.2.3 Classification

For the multi-class classification problem and the length of the feature vectors proposed, Discriminant Analysis classifier is expected to work well. Several classifiers were tested as detailed in Section 3.3.2. The Quadratic Discriminant Analysis (QDA) function, and the Linear Discriminant Analysis (LDA) function are similar in terms of the function and classification rules except the way covariance matrix is computed separately for each class (*i.e.*, varying, not identical). As a result, QDA tends to fit the data better than LDA.

3.3 Performance evaluation

In this section, we evaluate the performance of the proposed graph spectral features for in-air hand-drawn number and shape recognition. The experimental set up includes creation of a new dataset for both numbers and shapes using in-air hand-drawn by several users captured by a Kinect sensor as detailed in Section 3.3.1. The evaluation process includes evaluating the effect of different classifiers for the proposed features in order to find the optimal classifier, the effect of number of nodes, n, for graph formulation, the impact of the threshold value T and the choice

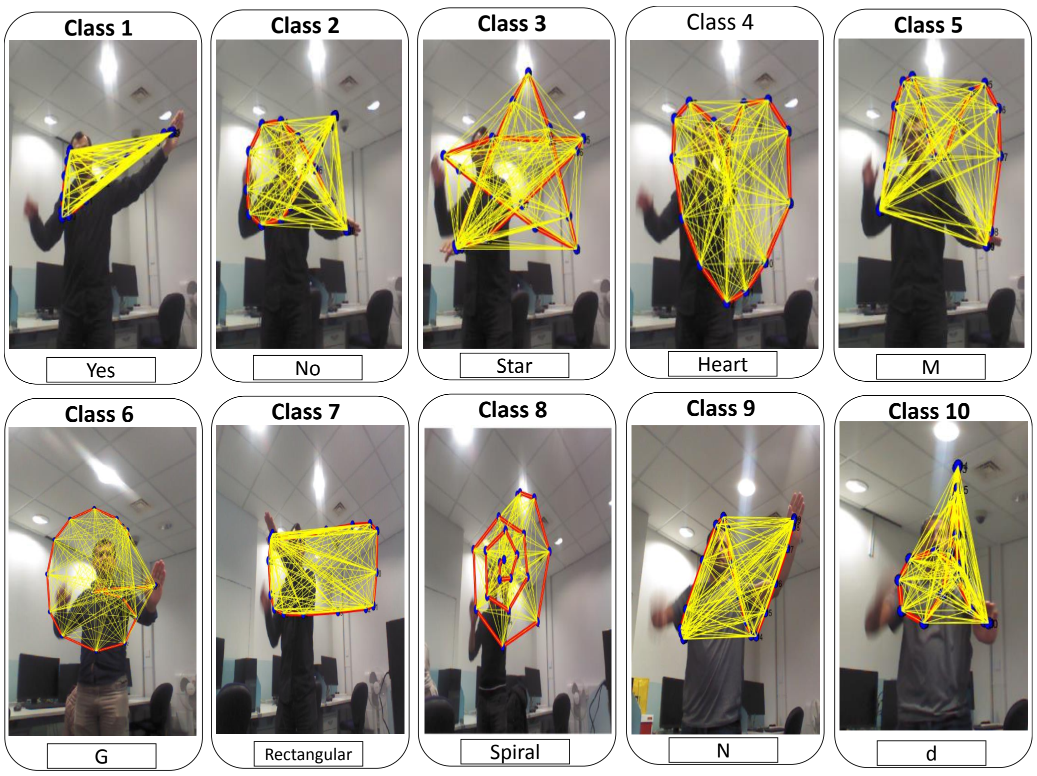


Figure 3.6: Ten classes of shape dataset with its graph construction.

of eigenvectors (from both normalised and combinatorial Laplacian matrices) for feature vector formation. The performances in terms of average recognition rates, confusion matrices and execution times are reported.

3.3.1 Dataset

The system we proposed in this chapter is designed for acquiring live data streams of the in-air hand-drawn numbers and shapes. However, the available datasets mostly consist of numbers recorded as images. The challenge in such cases is to identify the hand path, starting and ending points. We evaluate the proposed system using two different datasets, which provide a sequence of the hand writing (*i.e.*, hand path movement sequence is provided as a vector). The dataset presented in [54] consists of 1000 training samples and 2300 testing samples (**d1**), while the dataset presented in [60] consists of 100 samples per number (0-9) captured using a PrimeSense 3D camera (**d2**).

In order to evaluate our proposed method using a larger dataset, we have created a new in-air hand-drawn numbers and shapes dataset (**d3**), details of which can be found in [127]. The samples in our dataset, acquired using Kinect, provide the direct observation of a sequence of hand movements (*i.e.*, start and end points coordinates). The database can be divided into two parts: numbers (Figure. 3.1) and arbitrary shapes (Figure. 3.6). The number sub dataset includes 500 instances per each number 0 to 9, resulting in a total of 5000 samples of in air hand-drawn numbers. Similarly, the shape sub dataset also includes 500 samples per each shape of 10 different arbitrary 3D shapes, resulting in a total of 5000 samples. The total samples equals 10,000 instance and samples of the datasets can be seen in Figure. 3.7. Our numbers and shapes datasets provide X, Y, Z coordinates of the users writing hand movement. The dataset provides raw sampled hand path with original sampling rates and without any normalising nor smoothing, resulting in various numbers of data points within a given hand path. Our approach for pre-processing was outlined in Section 3.2.1. This dataset creation has received The University of Sheffield ethics approval under application 023005 granted on 19/10/2018. Table 3.1 shows a brief description of each dataset.

3.3.2 Evaluation of different classifiers

In order to test the entire dataset instead of random partitioning, a 10-fold cross-validation procedure is implemented to find the optimal classifier. Table 3.2 shows the mean accuracy for each classifier. QSVM records relatively lower degree of accuracy than the other classifiers, whereas the Quadratic QDA shows the highest level of recognition rate with more than 99%, while both CT and KNN provide nearly 99% level of accuracy. Thus, QDA is used for the rest of the experiments.

3.3.3 Evaluation of different number of graph nodes and threshold values

In this experiment, the pre-processing steps were repeated for various values of n. We start the test using $n = 3$, which is the minimum number of nodes to form the graphs. As can be seen in Figure. 3.8, it is clear that too few number of nodes (*i.e.*, $n \leq 10$ nodes) is not suitable for accurately representing the samples. The level of accuracy is then relatively stable using

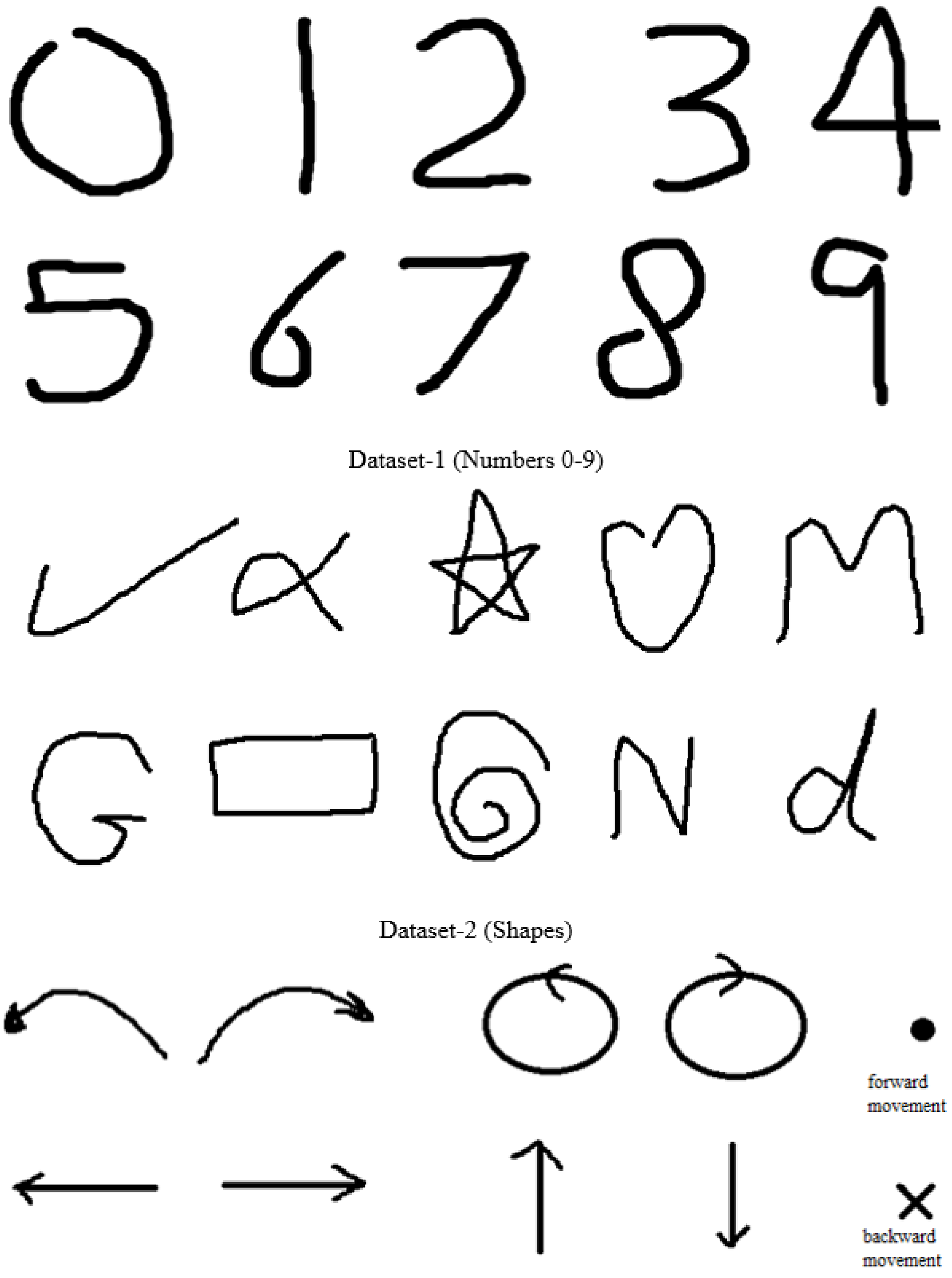


Figure 3.7: Samples of the proposed dataset [127], which can be divided into three datasets: numbers, shapes, and 3D gestures.

Table 3.1: Summary of the public datasets that can be used for the evaluation of the in-air hand-drawn shape recognition. tr and ts refer to the training and testing samples respectively.

Dataset	Reference	Type	Classes	Number of samples	Year
In-Air Hand-Drawn Number and Shape Dataset	[127]	Matlab & Text	Numbers& Shapes	$500 \times 20 = 10,000$	2018
Digit writing in the air	[54]	Text	Numbers	2300 (tr)+ 1000(ts)	2015
Gesture recognition	[60]	Text	Numbers	100×10	2013

Table 3.2: Average recognition rates (%) for different classifiers.

Classifier	QDA	QSVM	CT	KNN
Mean accuracy (%)	99.5	97.8	98.6	99.18

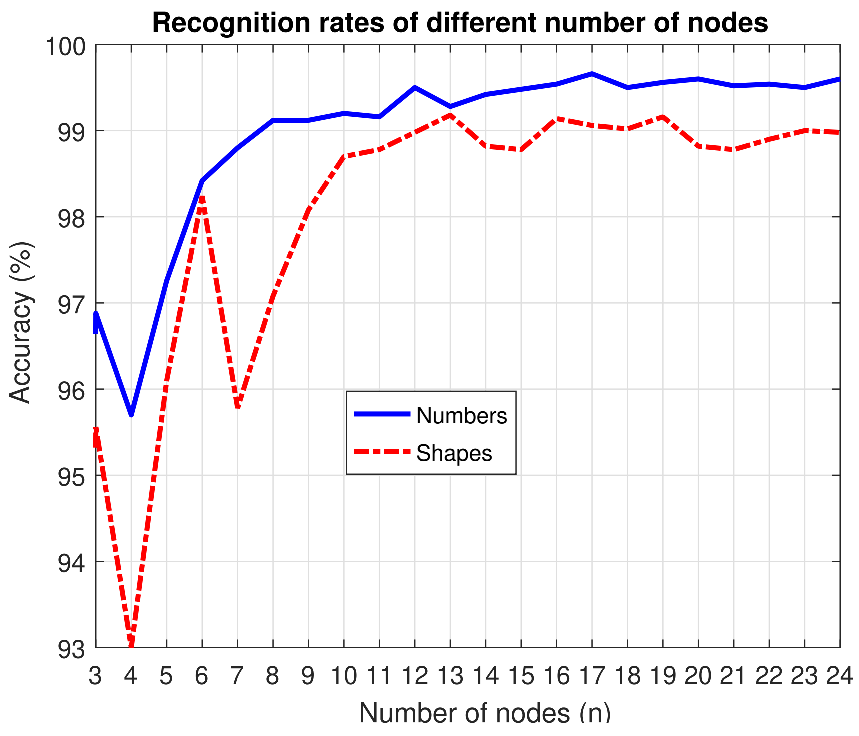


Figure 3.8: Recognition rates for different values of n at $T = 1$. In this experiments, we use the same set of feature but for different number of observation at a fixed threshold.

$12 \leq n \leq 24$ nodes. Note that 24 nodes is the minimum length among all samples in our dataset.

Similarly, the pre-processing steps were also repeated for various values of T. As can be seen in Figure. 3.9, it is evident that the optimal range lies in 1 to 8 unit distance. Then the accuracy falls dramatically using $T \geq 12$.

3.3.4 Evaluation of different eigenvectors

It has been shown that the geometric Laplacian matrix Γ has almost similar behaviour to the eigenvectors of normalised Laplacian matrix $\mathcal{L}$ [35]. Therefore, in this experiment, we only evaluate the individual eigenvectors from $\mathbf{L}$ and $\mathcal{L}$, as the feature vector. Figure. 3.10 shows the accuracy rate of using different graph eigenvector instead of $\mathbf{u}_0$ in Eq. (3.5), Eq. (3.7), and

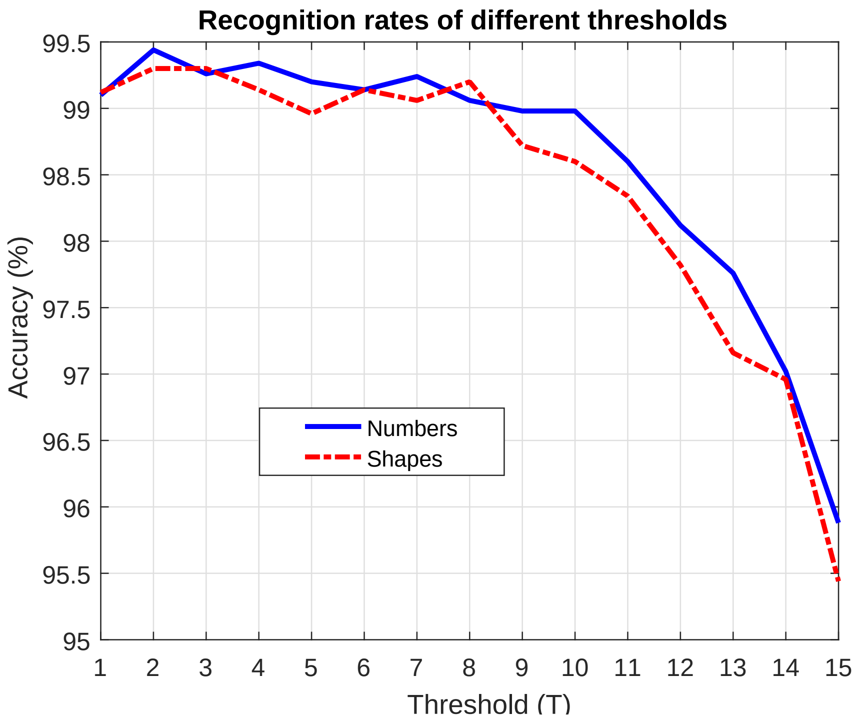


Figure 3.9: Recognition rates for different values of T at $n = 17$. In this experiments, we use the same set of feature but for different values of the threshold at a fixed length.

Eq. (3.8). It can be seen that the $\mathbf{u}_0$ of $\mathcal{L}$ provides the highest recognition rate of 99.56%, as it captures details of the structure of the graph. The second best result is from the $\mathbf{u}_{N-1}$ of $\mathcal{L}$, which corresponds to the maximum eigenvalue (λ_{max}). For most eigenvectors, those from $\mathcal{L}$ appear to outperforming those from $\mathbf{L}$.

3.3.5 Performance of the proposed method

From the above experiments, we use ($n = 17$, $m = 15$, $T = 2$) for numbers and ($n = 13$, $m = 15$, $T = 3$) for shapes to construct the graph. This is followed by using $\mathbf{u}_0$ of $\mathcal{L}$, for generating the feature vector components $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ and the QDA classifier to evaluate the performance of the proposed method using our dataset. The corresponding confusion matrices for numbers and shapes are shown in Figure. 3.11 and Figure. 3.12. The proposed method has achieved average recognition rates of 99.56% and 99.44% for numbers and shapes, respectively.

In order to compare the proposed method with the state-of-the-art methods in the literature, we first evaluate the method using an existing datasets in **d1** and **d2**. For the two datasets, the

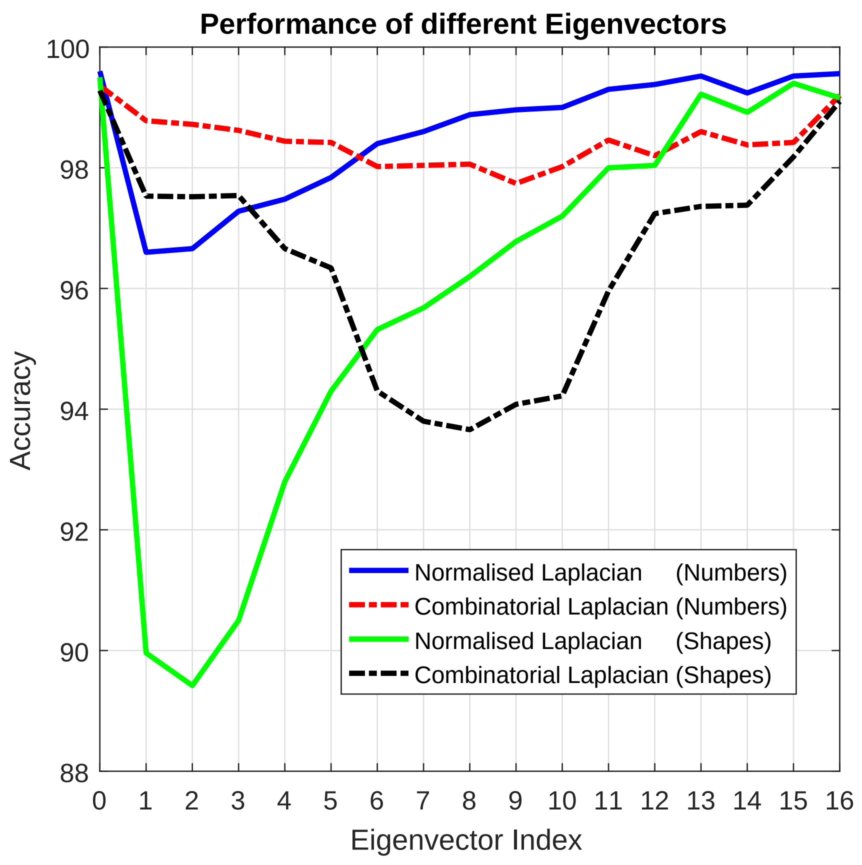
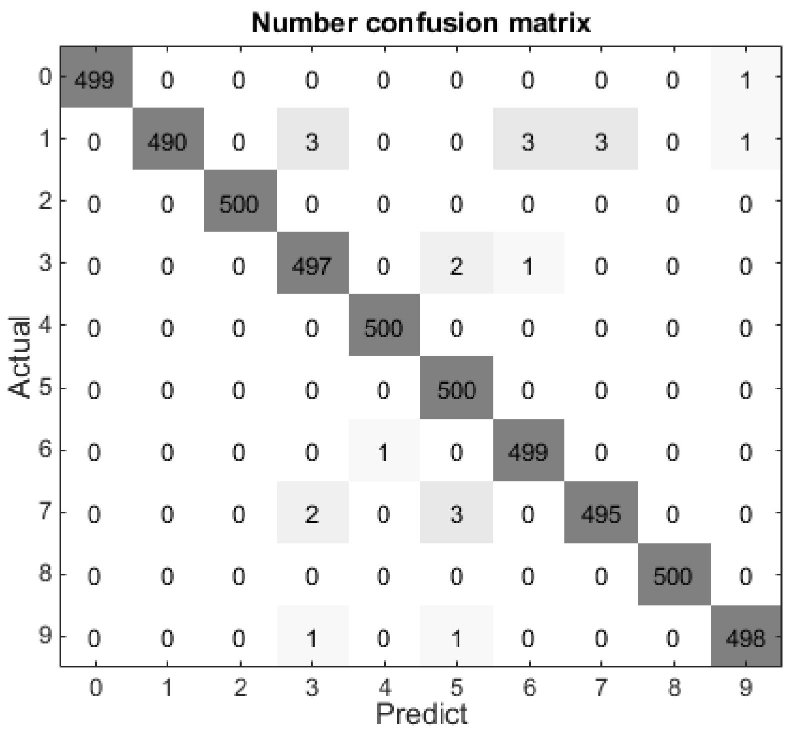
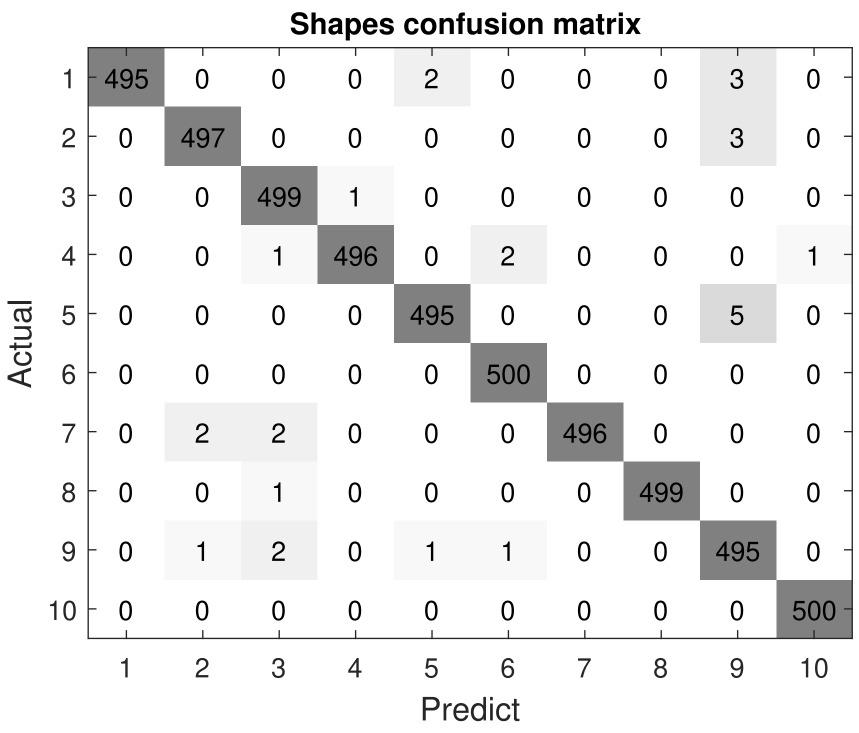


Figure 3.10: The accuracy rate of individual eigenvectors of the normalised and combinatorial graph Laplacian matrices using QDA classifier for both numbers and shapes. In this experiment, individual graph eigenvectors are used instead of $\mathbf{u}_0$ in Eq. (3.5), Eq. (3.7), and Eq. (3.8) to compute the features.



Actual \ Predict	0	1	2	3	4	5	6	7	8	9
0	499	0	0	0	0	0	0	0	0	1
1	0	490	0	3	0	0	3	3	0	1
2	0	0	500	0	0	0	0	0	0	0
3	0	0	0	497	0	2	1	0	0	0
4	0	0	0	0	500	0	0	0	0	0
5	0	0	0	0	0	500	0	0	0	0
6	0	0	0	0	1	0	499	0	0	0
7	0	0	0	2	0	3	0	495	0	0
8	0	0	0	0	0	0	0	0	500	0
9	0	0	0	1	0	1	0	0	0	498

Figure 3.11: Confusion matrix for individual numbers (Overall average recognition rate is 99.56%).



Actual \ Predict	1	2	3	4	5	6	7	8	9	10
1	495	0	0	0	2	0	0	0	3	0
2	0	497	0	0	0	0	0	0	3	0
3	0	0	499	1	0	0	0	0	0	0
4	0	0	1	496	0	2	0	0	0	1
5	0	0	0	0	495	0	0	0	5	0
6	0	0	0	0	0	500	0	0	0	0
7	0	2	2	0	0	0	496	0	0	0
8	0	0	1	0	0	0	0	499	0	0
9	0	1	2	0	1	1	0	0	495	0
10	0	0	0	0	0	0	0	0	0	500

Figure 3.12: Confusion matrix for individual shapes shown in Figure. 2.1 (Overall average recognition rate is 99.44%).

Table 3.3: Average recognition rate (%) for sign numbers.

Method	[55]	[53]	[54]	**Proposed method**			[60]	[126]
Dataset	**d1**	**d1**	**d1**	**d1**	**d3**	**d2**	**d2**	**d3**
Zero	92.61	97.83	97.83	97.82	99.8	100	98.5	99.62
One	76.09	94.78	91.3	93.47	98	98	99.5	93.85
Two	86.96	95.65	96.09	99.56	100	100	98.8	96.15
Three	86.96	93.48	97.39	96.95	99.4	99	99	99.62
Four	91.74	96.09	97.39	99.56	100	98	98.7	100
Five	75.96	89.13	98.26	95.21	100	100	97.5	94.23
Six	86.96	95.65	97.83	98.26	99.8	99	99	97.69
Seven	91.3	94.35	97.83	96.52	99	99	99	95.38
Eight	87.83	93.91	95.22	99.56	100	100	97.9	100
Nine	89.57	95.22	99.13	96.95	99.6	99	98.6	98.46
Average	90.8	94.6	96.82	**97.39**	**99.56**	**99.2**	98.63	97.53

Table 3.4: Average time requirement to perform different steps in the proposed system.

Step	Performance average time (ms)
Hand's path detection	2.729
Graph decomposition	0.055
Feature extraction	0.195
Classification	1.148
Full time system	4.127

parameters, ($n = 9$, $m = 35$, $T = 3$) and ($n = 24$, $m = 10$, $T = 2$) were used respectively for forming the graph. As can be seen in Table 3.3, this has achieved mean accuracy of 97.39% for **d1** and 99.2% for **d2**, which are both better than the methods in the literature that use those datasets [53–55, 60, 126]. Overall, the proposed method has achieved the best results compared to the existing methods in Table 3.3.

All experiments were implemented using Matlab R2015b on a PC with Intel processor, CPU@3.6GHz and RAM 16GB. The time requirement of operating number and shape recognition in a Kinect-based real-time system is about 4.127 ms per sample, which is suitable for a real-time in-air hand-drawn number and shape recognition system. The breakdown of the average times for components of the algorithm is shown in Table 3.4.

Since the graph adjacency matrix was defined based on the connectivity, it is insensitive to

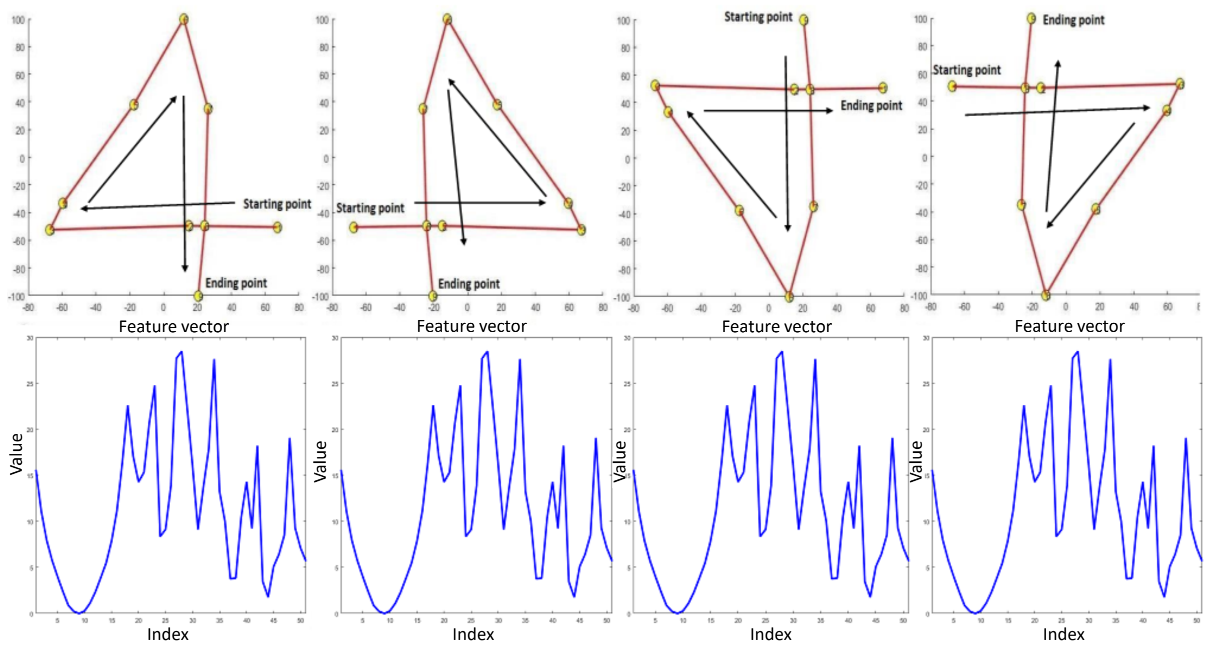


Figure 3.13: Hand drawing in different rotating angles (top row) with corresponding features (bottom row). We can see in the bottom row that the features are the same despite the drawing angle.

the rotation and flip changing. In other words, it does not matter what is the angle or direction of writing as long as it follows the rules of starting and ending point. For example, all the cases shown in (Figure. 3.13) are detected as number four. Due to this reason, the proposed method is invariant to the rotation or flipping or orientation angle variation of the in-air drawn samples. More example on the rotation invariant property can be seen on [126].

3.4 Concluding remarks

In this chapter, we have presented novel graph spectral features for in-air hand-drawn number and shape recognition. The proposed method includes pre-processing for converting the hand path movement, captured via Kinect, into a fully connected graph followed by analysis of the eigenvectors of the normalised Laplacian of the graph adjacency matrix for forming the feature vector. We have utilised the eigenvector, $\mathbf{u}_0$, as it captures the details of the structure of the graph. The proposed method has resulted in the highest performance with accuracies of 99.56% and 99.44%, for numbers and shapes, respectively, out performing the existing methods for three different datasets. The proposed method also has the added benefits of fast operation and invariance to rotation and flipping. In the next chapter, more complex shapes with high similarities will be considered in the recognition process.

Chapter 4

Graph spectral domain based on local details for shape recognition

4.1 Introduction

In the previous chapter, we solve the problem of matching low-level structures. In this chapter, we will explore the graph spectral features matching for the medium-level structures. Object recognition in terms of shape analysis has recently received a great attention in the field of computer vision [18] and applications, such as, security [131], video gaming [72], medical imaging [132] and human activity and pose understanding [133].

The detection of shape appearance, part-structure, occlusion, articulation, and local details play an important role in 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 [134]. For example, although, various shapes in Figure. 4.1 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

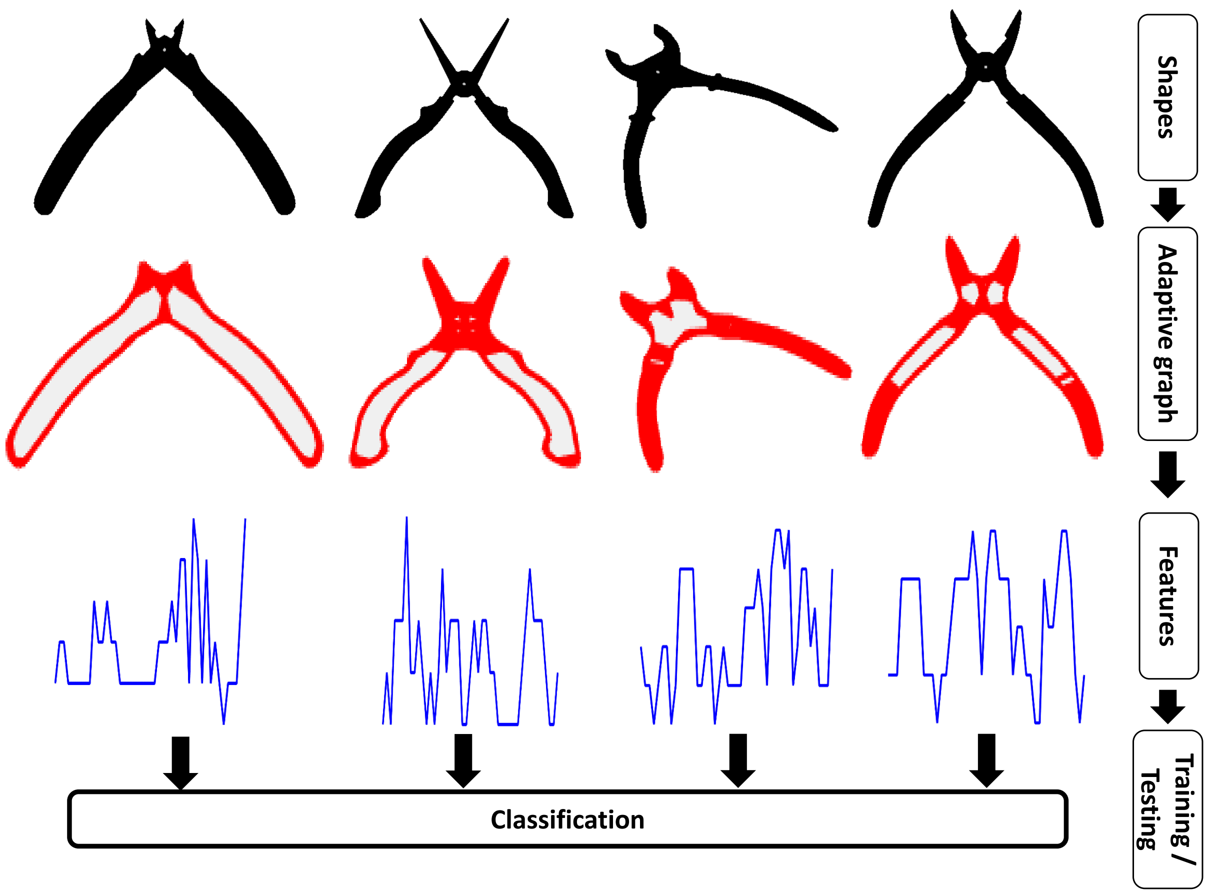


Figure 4.1: Challenging objects belong to four different classes that have high conceptual similarity structures (top row) and their associated adaptive graph connectivity. The top row shows how a human can see and the bottom row explains how a computer can see.

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 [25], inner-distance [135], complex-network [92], short-cut [18], part-alignment [72], and shape contexts [136]. These studies aimed to achieve the optimal representation of shapes by investigating on its outline. However, main limitations of these approaches are high computational complexity, restrictions on some shape sizes [25] and sensitivity to noise [135]. Although deep learning [62, 63] has achieved significant performance in object classification, such algorithms need extensive training data and time [137]. 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 [10, 138, 139]. 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 [12, 13]. More importantly, another study on human vision suggests that the visual cortex perceives and understands shapes be representing the shape boundary as a connected set of nodes [11], which has inspired us for the proposed method in this work.

In this study 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 an implicit threshold to build a graph from shape nodes to capture complex shape structures and details. A set of discriminating features is then extracted on graph spectral domain followed by classification using machine learning techniques. In the present chapter, we address representing both 2D and 3D shapes. By 3D shapes, we refer to the shapes perceived by point clouds of 3D objects. The main contributions of this work 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.

The rest of the chapter is structured as follows: a comprehensive explanation of the graph concepts, graph model based on adaptive connectivity and graph spectral features are shown in Section 4.2. Then, Section 4.3 evaluates the proposed system based on different classifiers and data-sets. Finally, the concluding remarks are shown in Section 4.4.

4.2 The proposed method

The proposed method can be summarised mainly into four steps as shown in Figure. 4.2. The framework begins by representing the shape's silhouettes using edge detector filter for 2D shapes and Growing Neural Gas (GNG) for 3D shapes. The resulted pixels are used as graph nodes. Graphs are then created by updating the allowable limits for node connections. After

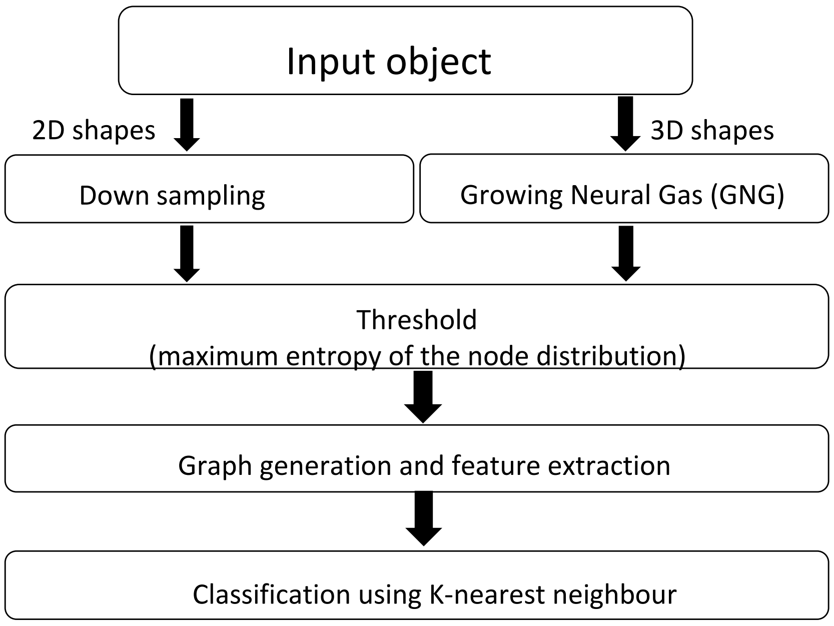

Input object
2D shapes
3D shapes
Down sampling
Growing Neural Gas (GNG)
Threshold
(maximum entropy of the node distribution)
Graph generation and feature extraction
Classification using K-nearest neighbour

Figure 4.2: The proposed method contains of four steps: shape representation, graph generation, feature extraction and classification.

getting the desired limit, a combination of node and spectral domain features are extracted to represent the shape. Lastly, these features are classified using a machine learning technique.

4.2.1 Shape representation

Since this work applies machine learning for classification, we need to specify a fixed number of pixels n to get the same length of features and reduce the complexity of 2D/3D shapes as follows:

2D Shape

Let $\mathbf{d} = (\ddot{\mathbf{H}}_\mathbf{k}, \mathbf{l}_\mathbf{k})$ be a dataset, where $\ddot{\mathbf{H}}_\mathbf{k}$ is the 2D binary shape and l is the corresponding label of sample k. For each $\ddot{\mathbf{H}}_\mathbf{k}$, we extract its contour (x, y) from the input image silhouette using an edge detector filter (*e.g.*, Sobel filter). The resulting path P is usually a smooth curve with N-pixels. Therefore, n-pixels are selected from P, where $n < N$, to create a uniform path $\hat{P}$ for graph generating as given in Eq. (3.1). $\hat{P}$ is then used to generate the graph.

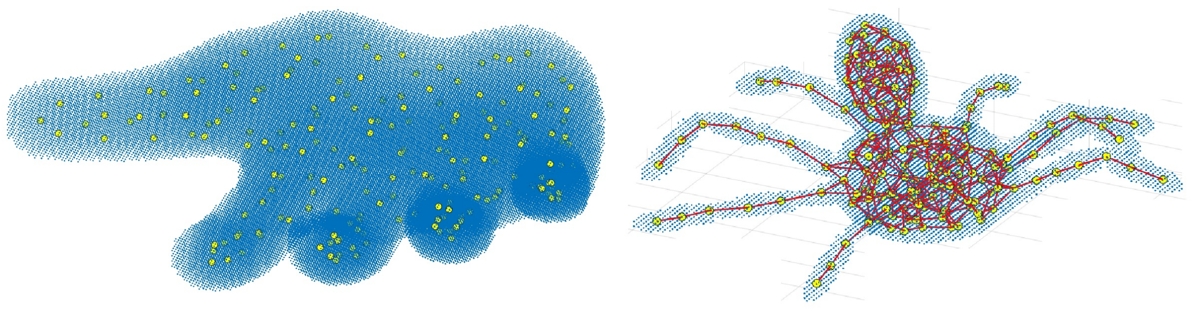


Figure 4.3: Example of applying the GNG on 3D shapes. Left side, we show only the nodes. right side, we show the node and its adaptive connectivity.

3D shape

While a silhouette border is extracted to represent the 2D shape, this is not possible for the 3D shape ($\dddot{\mathbf{H}}$) because it is represented as a surface. Therefore, we use (GNG) to get the same number of pixels to represent the samples. GNG is a simple unsupervised procedure to select the optimal pixels to represent the shape 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.

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.

In this study, the input data of GNG are the Cartesian coordinates of a 3D point cloud shape. Based on the Euclidean distance, pixels grow gradually inside the shape region during the training. At the end of training process, GNG should satisfactorily cover the shape regions as can be seen in Figure. 4.3. A detailed description of GNG can be found in [140] with its Matlab code. From GNG, we took only pixel locations and not their connections. The connection procedure is based on the proposed threshold. In the following subsections, we will provide information about: graph preliminary, graph connectivity, graph properties, threshold set up, feature extraction, and classification procedure.

4.2.2 Graph preliminaries

Suppose that $\mathcal{G}$ is an undirected graph, the $\mathbf{A}, \mathbf{D}, \hat{\mathbf{r}}$, and $\mathbf{L}$ are computed as in Eq. (2.1), Eq. (2.4), Eq. (3.2), Eq. (2.3). For $(\ddot{\mathbf{H}})$, z dimension is not considered in Eq. (3.2). We also define Φ_i as the number of edges incident on $\mathcal{V}_i$, which is also known as a node degree, as in Eq. (2.2). The following subsections will highlight the most relevant properties of graph spectral domain.

Graph connectivity

Connecting pixels is a key concept in generating graphs because the way to connect vertices has a direct effect on the spectral characteristics of the graph. The graph connectivity can be categorised into four types:

1. Special connectivity: for specific applications, vertices have their own connectivity without the ability to change it. An example of this is the Minnesota graph, where the edges represent the road network [35].

2. Full connectivity: when $\mathcal{V}_i$ is connected to all other vertices in the graph and each vertex has $(n-1)$ connections. This type of connectivity provides an efficient characterisation of global shapes.

3. K-Nearest Neighbour: where $\mathcal{V}_i$ is linked to the nearest K-vertices, and each vertex has K connections [141]. This type of connectivity is usually applied in a uniform grid such as image-based applications.

4. Conditional or circular connectivity: vertices are connected if a certain condition t is accomplished.
   $\mathcal{V}_i$ is connected to $\mathcal{V}_j$, if and only if $\mathcal{V}_j$ satisfies the requirement of a particular condition Figure. 4.4. Using this connectivity type, there is no fixed number of connections at each vertex and the number of connected elements depends on the condition.

Although a fully connected graph provides an efficient representation of the global shape as we showed in **Chapter 3**, the major drawback of this type is that local details are not reliably detected compared to the global shape. Since we aim to classify more complex shapes, this work uses a conditional connectivity to reveal the local information.

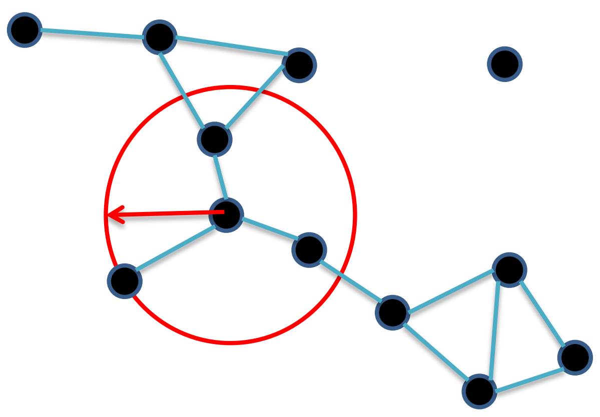


Figure 4.4: Conditional connectivity for a random graph. As we can see, vertices with the specified distance are connected, otherwise, they are not connected.

As mentioned earlier, the edges between the vertices represent the Euclidean distance of the pixels. Therefore, the proposed conditional connectivity means that each vertex is connected to other vertices that fall in less than a certain distance. Figure. 4.5 shows an example of how vertices are connected using different thresholds with their corresponding node degree (Φ). This type of connectivity can be used to reveal the protrusions on the shape's surface.

Graph properties.

In order to explain the effect of the conditional connectivity on the graph basis, Figure. 4.6 shows different graph generation using four threshold values, which are 6, 7, 8 and 9 unit pixels in A, B, C, and D respectively.

The corresponding graph eigenvalues are listed below:

$\lambda_{(t(A),n)} = [0,\ 0,\ 0,\ 0, ..., 0.50,\ 1.86, ..., 24.97]$.

$\lambda_{(t(B),n)} = [0,\ 0,\ 0,\ 0,\ 0.01,\ 0.06, ..., 34.54]$.

$\lambda_{(t(C),n)} = [0,\ 0.030,\ 0.050, ..., 49.94]$.

$\lambda_{(t(D),n)} = [0,\ 0.031,\ 0.054, ..., 60.49]$.

The graph eigenvalues reveal important properties of the shape characteristic and they can be interpreted for the shape matching process. These properties include information about the size and number of clusters as shown below:

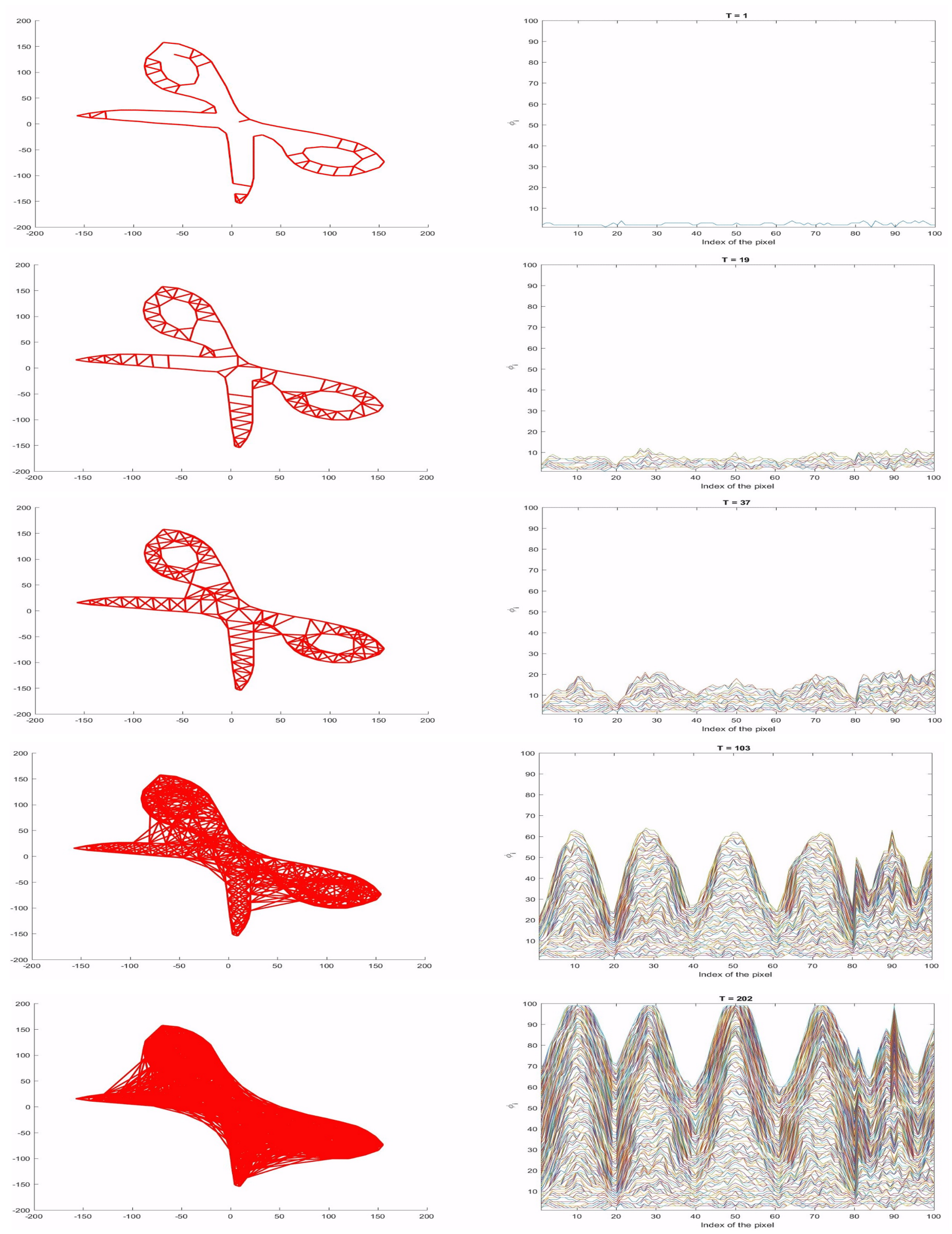


Figure 4.5: Different connectivity level with its corresponding node degree. In this experiment, we show five different levels of connectivity from a small distance to a larger distance. We can see how the connection grows with the threshold value.

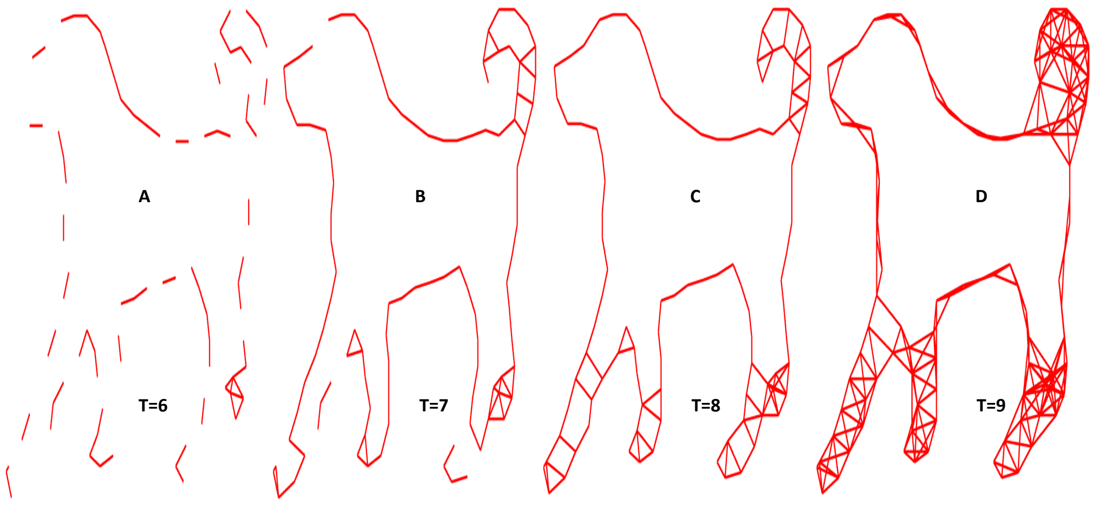


Figure 4.6: Graph construction using different threshold values. We can see how the connection grows with the increasing threshold value.

1. The shape density:

   Relatively, we can measure the density of the graph connectivity through the eigenvalues and, to be more accurate, through its last values $\lambda_{(n-1)}$. In this example for the same shape, we can see that $\lambda_{(n-1)}$ reflects the intensity of the graph connectivity. Also, it is always that

   $$\lambda_{(t,n-1)} < \lambda_{(t+1,n-1)}, \tag{4.1}$$

   which means that the last eigenvalue is always directly proportional to the value of t.

   $$\lambda_{(n-1)} \ \alpha \ t. \tag{4.2}$$

   This property is used to interpret the shape density.

2. The number of clusters (ω).

   Since we used a distance as a threshold to link vertices, in some cases, there may be some pixels without links or a group of nodes separated as one group. This can be detected using the number of zeros in the eigenvalues of the graph [35], which means that the eigenvalues of the graph can be used as an indicator of the number of clusters in the shape. The number of existing clusters depends on the threshold and it is calculated as shown in Eq. (4.3) and Eq. (4.4) [35].

$$\omega = \sum_{i=0}^{n-1} b_i, \tag{4.3}$$

where,

$$b_i = \begin{cases} 1, & \text{if } \lambda_i = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

For more clarification, Figure. 4.6A shows a dog with 39 groups and these group can be the sum of pixels or individual pixels. This is clearly shown in the number of zeros of the graph eigenvalues, where $\lambda_{(t(A),0\to 38)}$ are equal to zeros. The same thing in shape B, $\lambda_{(t(B),0\to 3)}$ are equal to zeros, which means that there are four groups ($\omega = 4$). For both shapes C and D, only λ_0 is equal to zero ($\omega = 1$), and that means all the existing pixels are connected as one group. In the rest of this work, ω will be used to state the graph connectivity and we will refer to the threshold value that makes $\omega = 1$ as $t_\circ$.

3. Spectral response:

   The behaviour of the graph eigenvectors of the Laplacian matrix depends on the connectivity. For example, if $\omega > 1$, the spectral response will be a pulse signal. However, if $\omega = 1$, the spectral response will be a sine wave as we can see in Figure. 4.7. In addition, we can see that the details of the object begin to appear in detail with the increased value of the condition. This property is vital in designing the adaptive graph because it gives a relative representation of the node connectivity.

4. The impact of n on the connectivity:

   $t_\circ$ is affected by the number of nodes (n). More pixels observed from the shape leads to the desired threshold $t_\circ$ being smaller and the opposite is also true. Thus, we can conclude that $t_\circ$ has an inversely proportional relationship with the number of observed pixels Eq. (4.5).

$$t_\circ \ \alpha \ \frac{1}{n}. \tag{4.5}$$

   Figure. 4.8 demonstrates the relation between n and $t_\circ$ using the same sample with $n = 40$

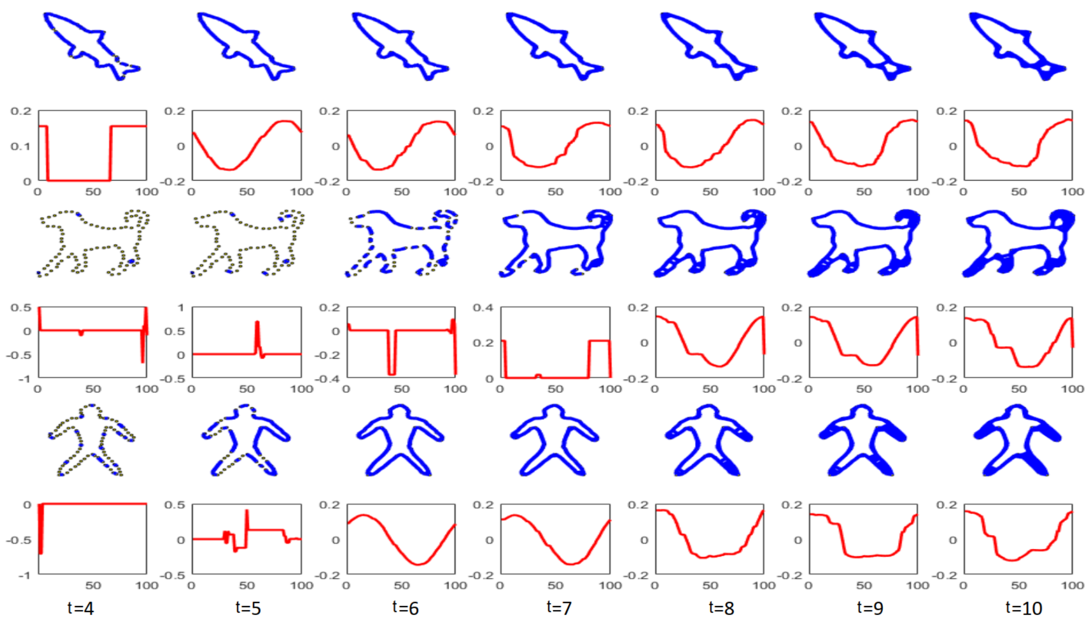


Figure 4.7: The spectral response (*e.g.*, Fielder vector) of the graph Laplacian matrix through different threshold values using three different shapes. In this experiment, we increased the threshold value from 4 to 10 pixels, and the result is a pulse or sine signal based on the shape topology.

and $n = 400$.

The information above highlight the importance of the graph spectral domain for shape understanding. We seek to generalise the method by considering any number of observations. To do this, two cases that lack local details should be considered as will be shown in Section 4.2.2.

Threshold set up

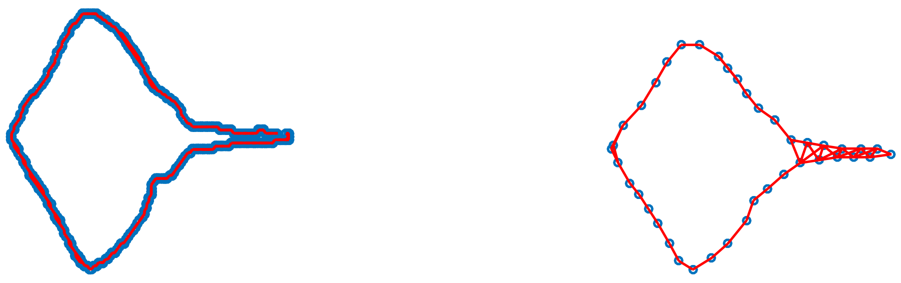
To achieve the optimal threshold, two rules have to be satisfied:

1. Lower boundary

   All the pixels must be linked as one group, which means that $\omega = 1$. Any threshold below this limit is not accepted.

2. Higher boundary

   It is advisable to increase the connection for a particular range δ to further highlight the local details as shown in Figure. 4.6 D. However, full connectivity, where each pixel has $(n - 1)$ connections, should be avoided because local details will be missing.



(a) $n = 400, t_\circ = 2$.

(b) $n = 40, t_\circ = 12$.

Figure 4.8: Two cases with big and small n (left column). We can see that t_0 is inversely proportional to n.

To summarise, $t_\circ$ is the minimum accepted distance but not always the optimal level. This is certainly true for the shape shown in Figure. 4.8a. This situation indicates the need for pixel down-sampling or increasing the threshold because pixels are only connected to their neighbours using $t_\circ$. Therefore, the threshold is updated by a fixed increment, and we can see that Φ begins with a mostly uniform number of connections, where most of the pixels are only connected to their neighbours. The local details at this level cannot be revealed. Then, these details start to appear after increasing δ as can be seen in the middle level of the Figure. 4.8a. Lastly, Φ then is fixed at $n-1$ level towards the maximum value of δ. Figure. 4.8b provides the same notion of Figure. 4.8a, except that the initial levels at $t_\circ$ are different. The constraint of Φ is explained in Figure. 4.9.

For all the samples, therefore, the optimal threshold T is computed by updating the initial $t_\circ$ with a specific dynamic range δ as shown in Eq. (4.6),

$$T = t_\circ + \delta. \tag{4.6}$$

The optimal threshold (T) is defined as the value that reveals the maximum details of the shape architecture. Therefore, for each object $(\ddot{\mathbf{H}})$ or $(\dddot{\mathbf{H}})$ with n nodes in $\mathbf{d}$ that contains k shapes, we compute the entropy of the normalised node distribution at $t_\circ$. Then, we increase δ from $0 \rightarrow (n-1)$ to find the node distribution corresponding to the maximum entropy as follows:

$$T = \underset{\delta}{\operatorname{argmax}}(\boldsymbol{E}), \tag{4.7}$$

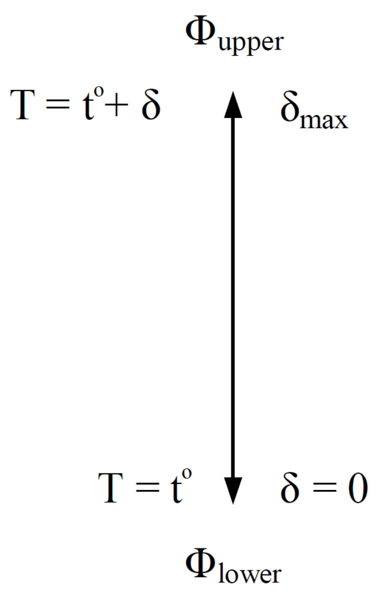


Figure 4.9: The upper and lower boundaries of Φ.

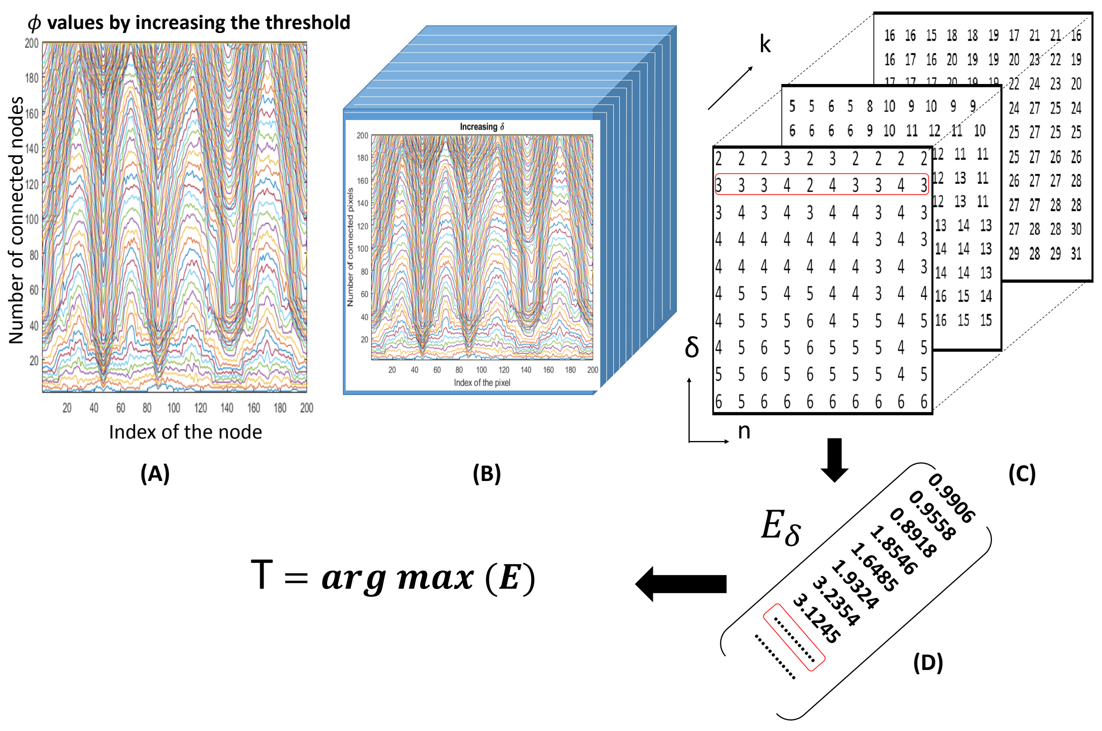


Figure 4.10: The dynamic evaluations, A: number of connected pixels for one sample, B: combining all samples, C: matrix representation of the connectivity for all samples in dataset, D: the entropy of each ϕ.

Algorithm 1 Input = ($\ddot{\mathrm{H}}$) or ($\dddot{\mathrm{H}}$), Output = (T)

```
 1: for $i = 0 : (k-1)$ do
 2:     $\hat{\boldsymbol{P}}_{\boldsymbol{i}} \leftarrow$ Shape representation as ($x_i$,$y_i$,$z_i$).
 3:     $t_\circ \leftarrow$ Threshold at $\omega = 1$.
 4:     for $\delta = 0 : (n-1)$ do
 5:         $\boldsymbol{\Phi_\delta} \leftarrow$ Node distribution at ($\delta + t_\circ$).
 6:         $\hat{\boldsymbol{\Phi_\delta}} \leftarrow$ Normalising ($\frac{\boldsymbol{\Phi_\delta}}{max(\Phi_\delta)}$) .
 7:         $E_\delta \leftarrow$ Compute the Entropy of $\hat{\boldsymbol{\Phi_\delta}}$.
 8:     end
 9:     $T_i \leftarrow max(\boldsymbol{E})$.
10: end
```

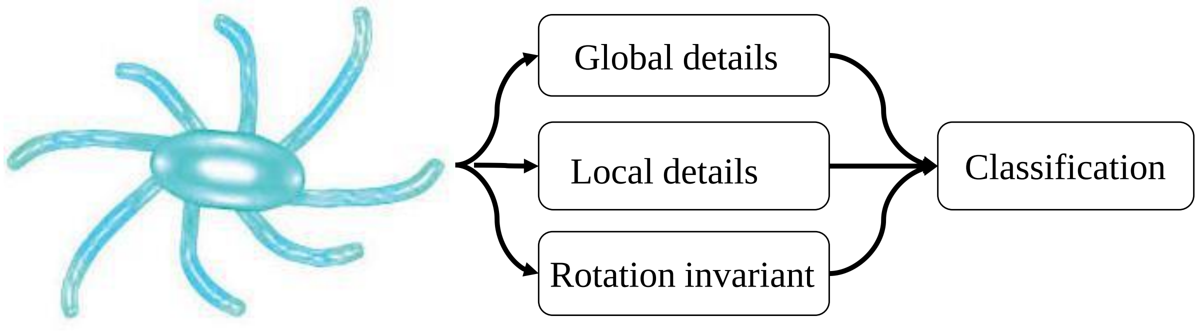


Figure 4.11: Three types of features are extracted to form a shape signature.

where,

$$\boldsymbol{E} = \sum_{i=0}^{n-1} \boldsymbol{\Phi_i} \log_2 \left(\frac{1}{\Phi_i} \right) . \tag{4.8}$$

The procedure of computing the proposed threshold of all samples in dataset is illustrated in Algorithm 1 and Figure. 4.10. A short demo can be found in [142].

4.2.3 Graph Spectral Features (GSF)

In order to classify samples, it is important to create a unique signature for each class using the prior interpretation of the adaptive graph. To achieve such a task, the proposed features should detect the local and global details of the shape. In addition, they should also be invariant to the rotation changes (Figure. 4.11).

For a given object, we calculate its features $\mathrm{GSF}_k = \{f_a, f_b, f_c\}$ as follow:

1. $$f_a = \hat{\mathbf{r}}_i \, \lambda_{(i)}, \qquad\qquad i = 0, \ldots, n-1. \tag{4.9}$$

   f_a is the scalar product between the eigenvalue λ_i and the corresponding distance $\hat{f}_i$. This provides an efficient descriptor to estimate the global structure and the density of the connectivity as described earlier in the example shown in Figure. 4.6. Figure. 4.12 and Figure. 4.13 show the average eigenvalue at each node for different classes with its standard deviation. The eigenvalues provide a clear distinction between various classes of four datasets.

2. $$f_b = \hat{\boldsymbol{\Phi}}_{\boldsymbol{i}}, \qquad\qquad i = 0, \ldots, n-1. \tag{4.10}$$

   Node distributions f_b is used at a certain level, which gives the maximum entropy to reveal the local details of the shape's surface.

3. f_c :

   In addition, we include a variety of other statistics of the node distribution for the detection process. This information is not affected by changes in shape rotation; such as:

   (a) $f_{(c,1)}$ = the mean.

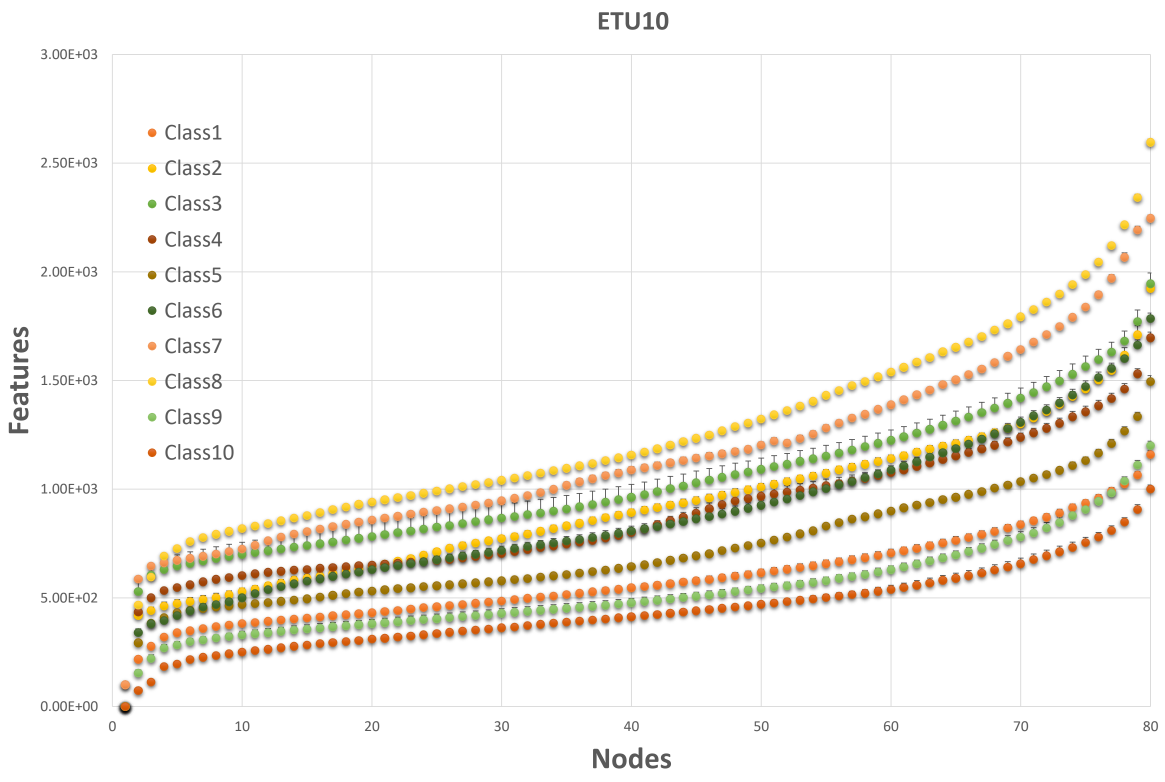
   (b) $f_{(c,2)}$ = the variance.

   (c) $f_{(c,3)}$ = the entropy (Eq. (4.8)).

   (d) $f_{(c,4)}$ = the summation of square node distribution Eq. (4.11).

$$f_{(c,4)} = \sqrt{\sum_{i=0}^{n-1} \hat{\boldsymbol{\Phi}}_{\boldsymbol{i}}^{\mathbf{2}}}. \tag{4.11}$$

Concatenating all features leads to an effective representation of shapes, which are classified in the next step using Machine learning principles. The total length of the features is (2n + 4).



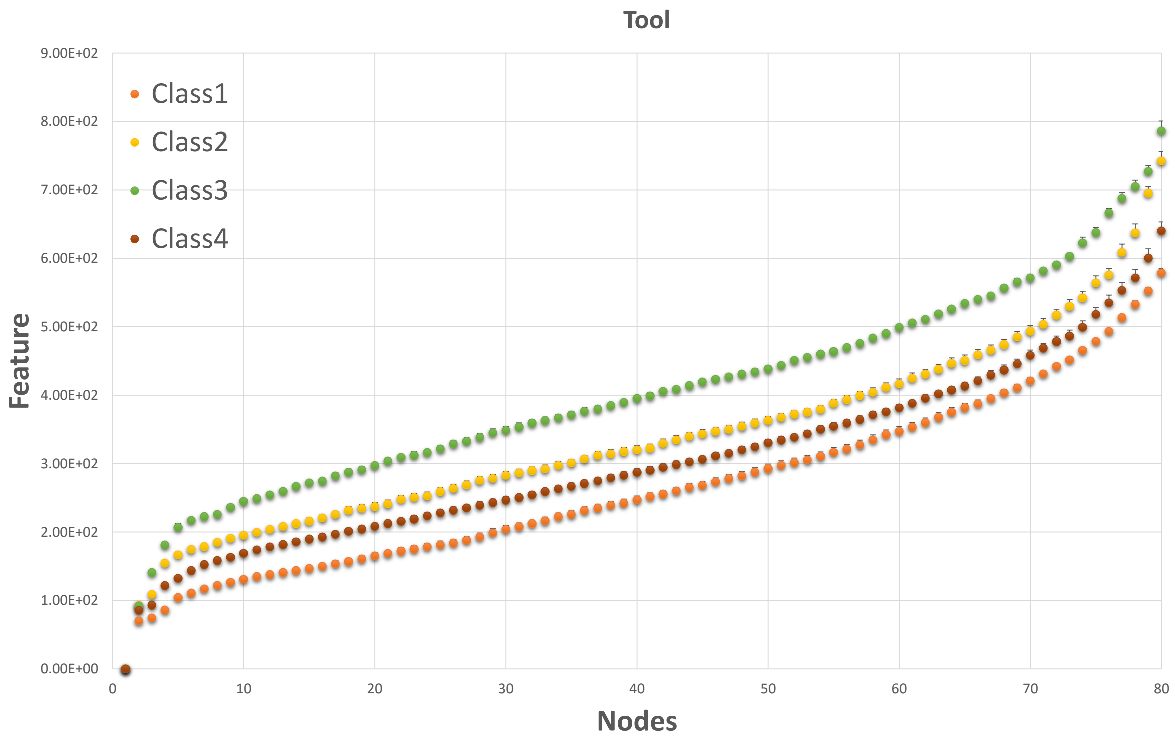
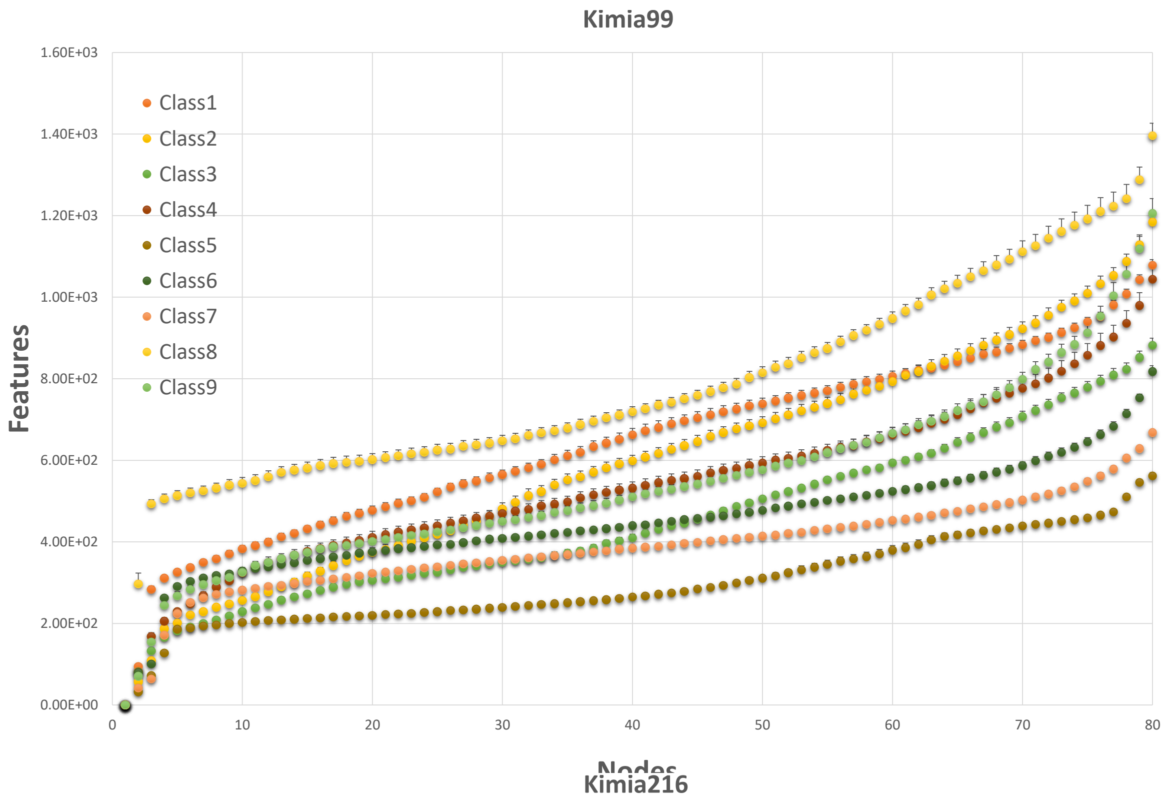


Figure 4.12: The f_a (y-axis) at each node (x-axis) for ETU10 and Tool datasets. Colours are corresponding to different classes.



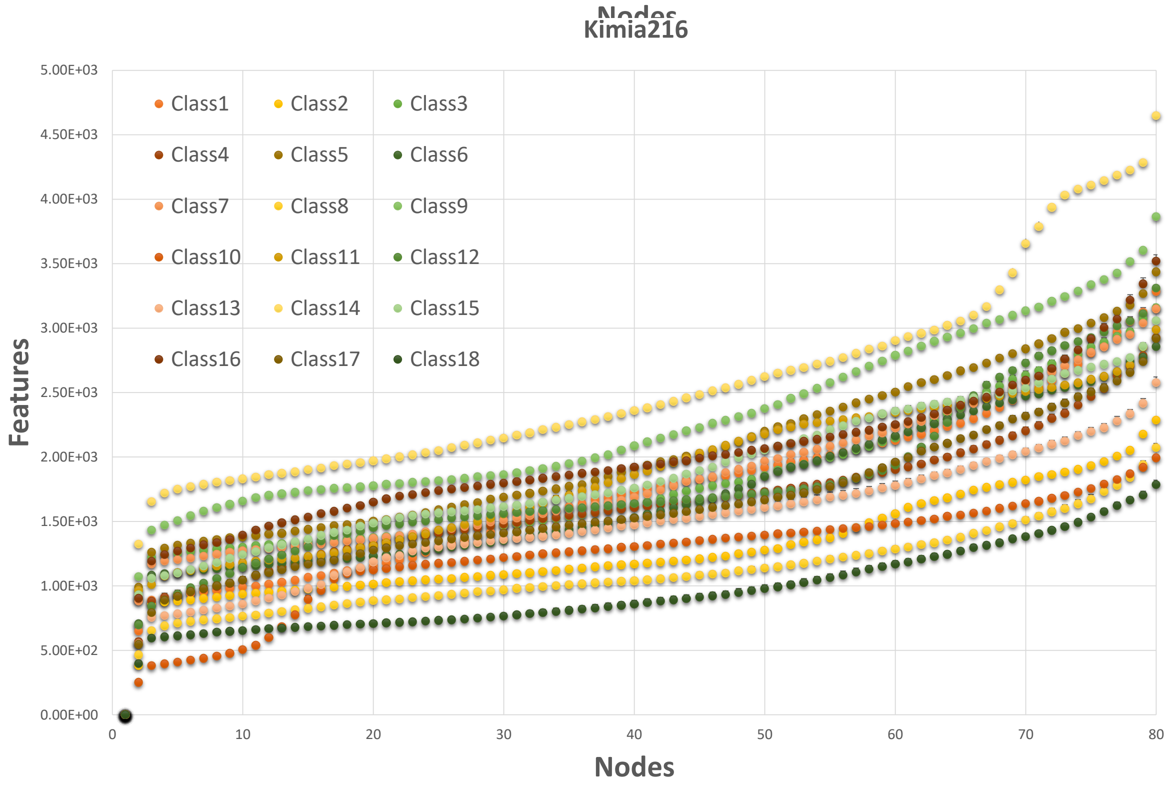


Figure 4.13: The f_a (y-axis) at each node (x-axis) for kimia99 and Kimia216 datasets. Colours are corresponding to different classes.

4.2.4 Machine learning

Based on several experiments conducted to select the optimal classifier, Nearest Neighbour (KNN) shows better performance compared to other classifiers in terms of accuracy and time processing, as will be shown in Section 4.3. The experiments include using SVM, KNN, CT, QDA, NN.

4.3 Performance evaluation

This section describes the performance evaluation of the proposed method for 2D/3D shapes classification. All the experiments are implemented using MATLAB R2018a on a PC with Intel 3.6 GHz processor and 16 GB RAM. Here, we will start by introducing the datasets, which are used in the evaluation process:

4.3.1 Datasets

We are keen to test the proposed method against a variety of shapes, sizes, and datasets, which vary in orientation, articulation, and scales. Thus, a large number of experiments are performed using four 2D shape datasets and two 3D shape datasets as follows:

1. ETU10 silhouette dataset (**d4**).
   One of the most well-known 2D databases is the EUT10 database [143], which provides a 5-degree rotation difference for each class. This dataset has 10 classes $\times$72 shapes in each class = 720 total images. Sample silhouettes from each class are shown in the top two rows of Figure. 4.14A. The bottom row shows different angles of the object. The ten classes in the confusion matrix correspond to the Bed, Bird, Fish, Guitar, Hammer, Horse, Sink, Teddy, Television and Toilet respectively.

2. Tool dataset (**d5**).
   The tool dataset [144] is one of the most challenging dataset as it has a conceptual similarity in its shapes. It consists of 35 articulated shapes, which are classified into four classes: 10 scissors, 15 pliers, 5 knives and 5 pincers respectively as shown in Figure. 4.14B.

3. Kimia 99 dataset (**d6**).

   The Kimia 99 dataset [145] consists of 9 classes $\times$ 11 samples = 99 images as shown in Figure. 4.14C. The nine classes in Tool dataset correspond to the Fish, Hand, Human, Aeroplane, Ray, Rabbit, Misk, Spanner and Dog respectively.

4. Kimia 216 dataset (**d7**).

   The Kimia 216 dataset [146] consists of 18 classes $\times$ 12 samples = 216 images as shown in Figure. 4.14D. The 18 classes in the confusion matrix correspond to the Bird, Bone, Brick, Camel, Car, Children, Classic, Elephant, Face, Fork, Fountain, Glas, Hammer, Heart, Key, Misk, Ray and Turtle respectively.

5. SHERC2010 dataset (**d8**).

   SHREC2010 dataset [147] consists of 20 objects $\times$ 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 Figure. 4.15. The classes include: Ants, Crabs, Hands, Humans, Octopus, Pliers, Snakes, Spectacles, Spiders, and Teddy respectively.

6. 3D shape benchmark dataset (**d9**).

   3D shape benchmark dataset [148]. This dataset consists of 19 classes$\times$ 20 samples per class $= 380$ shapes in total. Objects were presented in different orientations, scales and articulation, and that makes it one of the most challenging datasets. 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 objects are shown in Figure. 4.15.

4.3.2 Classifiers

Initially, we test different classifiers in terms of speed and the score for the recognition process. A 10-fold cross validation scheme is utilised to train and test all the samples in the dataset. Table 4.1 shows the accuracy of all the 2D datasets. As we can see, NN and KNN show the highest level of accuracy among all the classifiers. However, KNN classifier is much faster than NN. Therefore, for the rest of the experiment, we select KNN for the classification process.

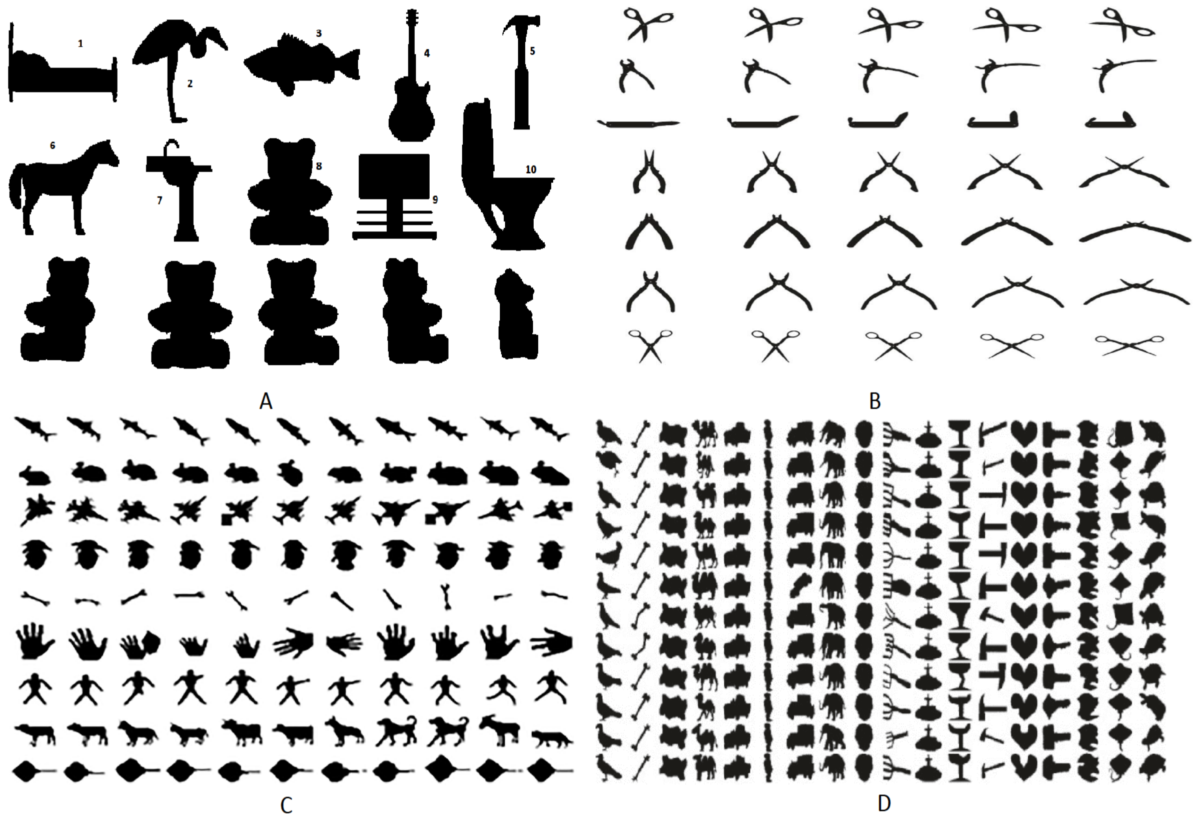


Figure 4.14: 2D datasets, which are used in this section, include A: ETU10 silhouette Dataset, B: Tools Dataset, C: Kiama99 dataset and D: Kiama216 dataset.

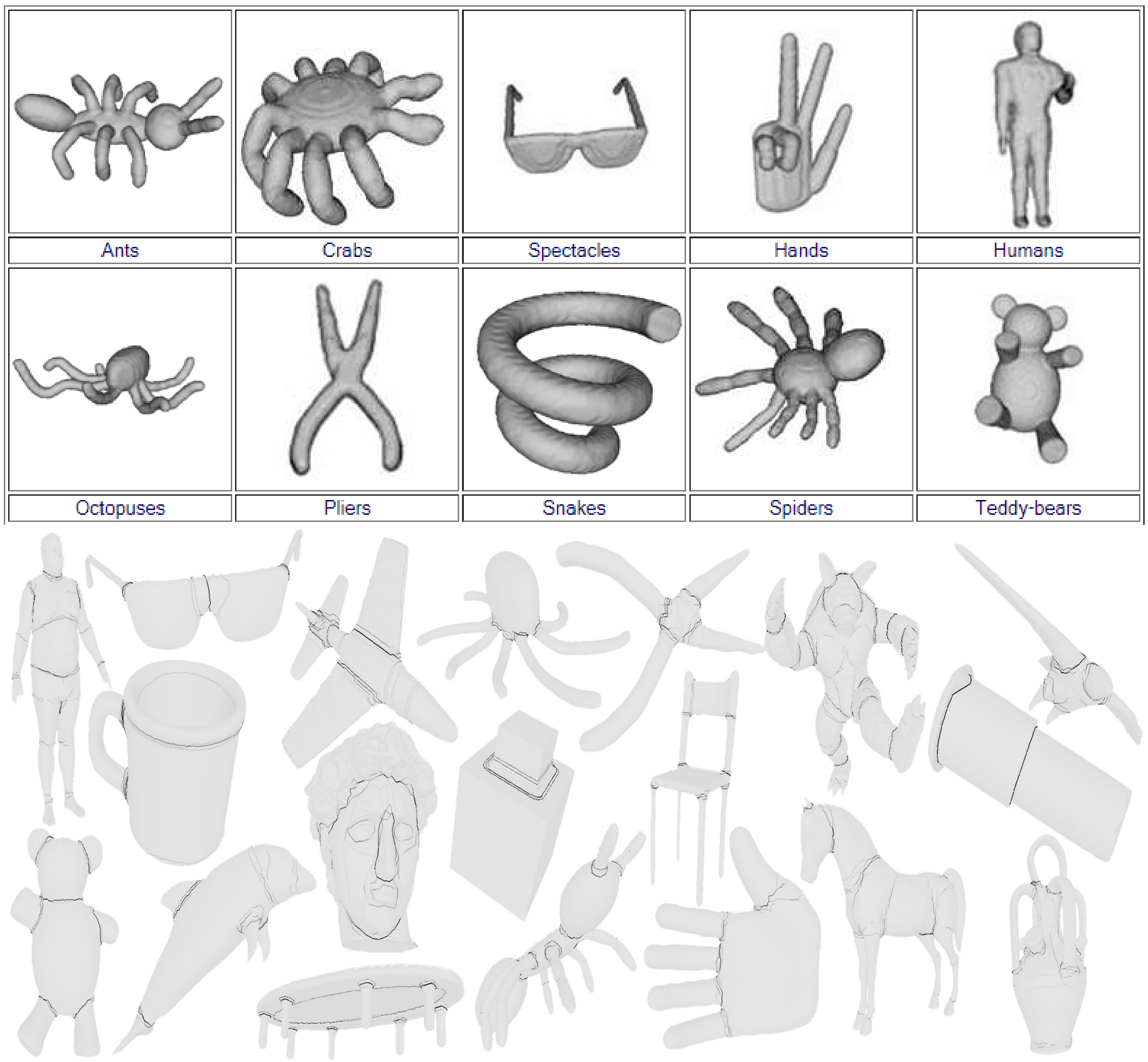


Figure 4.15: Samples of SHREC2010 dataset (Top), and 3D shape benchmark dataset (Bottom).

Table 4.1: Recognition score (%) using different classifiers and datasets.

Dataset	CSVM	KNN	CT	DA	NN
d4	99.58	99.58	94.18	89.54	99.7
d5	100	100	92.85	100	100
d6	97.98	100	90.82	91.91	99
d7	95.37	96.30	89.88	90.27	96.8

KNN classifier recognises samples based on the available trained classes (*i.e.*, KNN is a class-based classifier rather than individual samples). Several types of distances are tested such as Euclidean, Manhattan, Cosine, Mahalanobis, Correlation, Minkowski, Standardised Euclidean, and we found that the normal Euclidean distance provides an efficient performance.

4.3.3 Confusion matrices

In this study, we will use the confusion matrix to show the classification results of the individual dataset.

- For both **d5** and **d6**, samples are classified correctly with 100% accuracy. Although Tool dataset provides high similarity structures, the proposed method forms significant descriptors for the individual class.
- Despite the different angles of the samples in **d4**, the proposed features recognise samples with a high level of accuracy exceeding the state-of-the-art performance by 99.58%. These samples are classified without any confusion samples as we can see in Figure. 4.16.
- **d7** is one of the most challenging dataset because of the small number of available samples in each class compared to the total number of classes. This leads to a challenging score of up to 96.3% as shown in Figure. 4.16.
- As shown in Figure. 4.16, the proposed GSF recognises all models in **d8** with a high accuracy score of 93%. The only confusing sample is the octopus, which has been matched with spiders and this is because of the similarity in size (ie eigenvalues) and structure.
- **d9** has been classified with an accuracy level of (76.32%) as shown in Figure. 4.16. This dataset has more misclassified samples compared to other such as in Bearing, Vase, Four

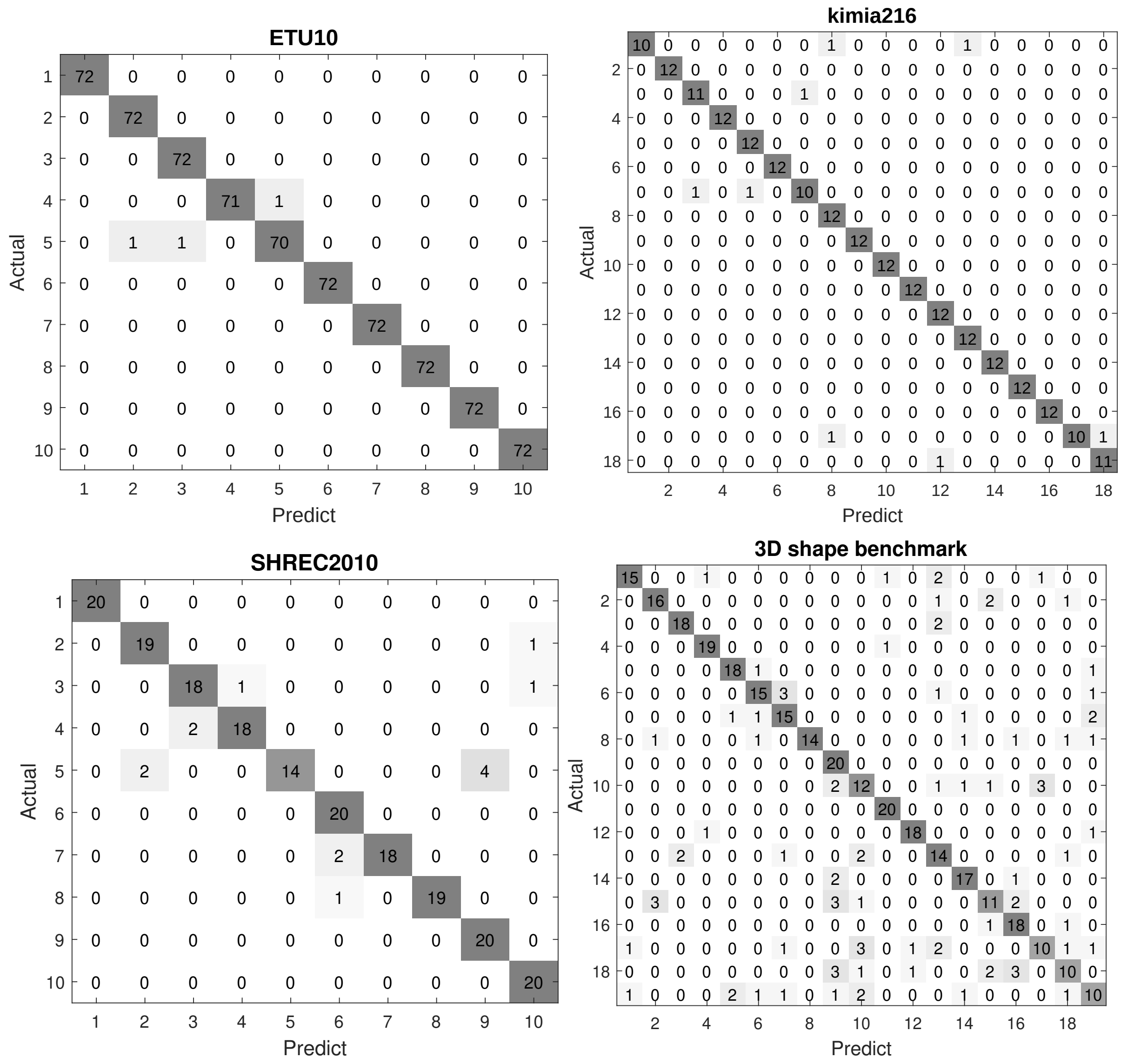


Figure 4.16: Confusion matrices of different datasets, which are described in Section 4.3.1.

Leg, and Bust. This is because the same object appears in different poses and articulation.

4.3.4 The proposed threshold versus number of observation

Figure. 4.8 illustrates that, for the same shape, the threshold value depends on the number of observations. In order to provide a general idea about the system performance using a wide range of n and T, this study uses Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [149]. The TOPSIS is an optimisation method, which helps to understand and analyse the relationship between multi-parameters based on cost function. In this experiment,

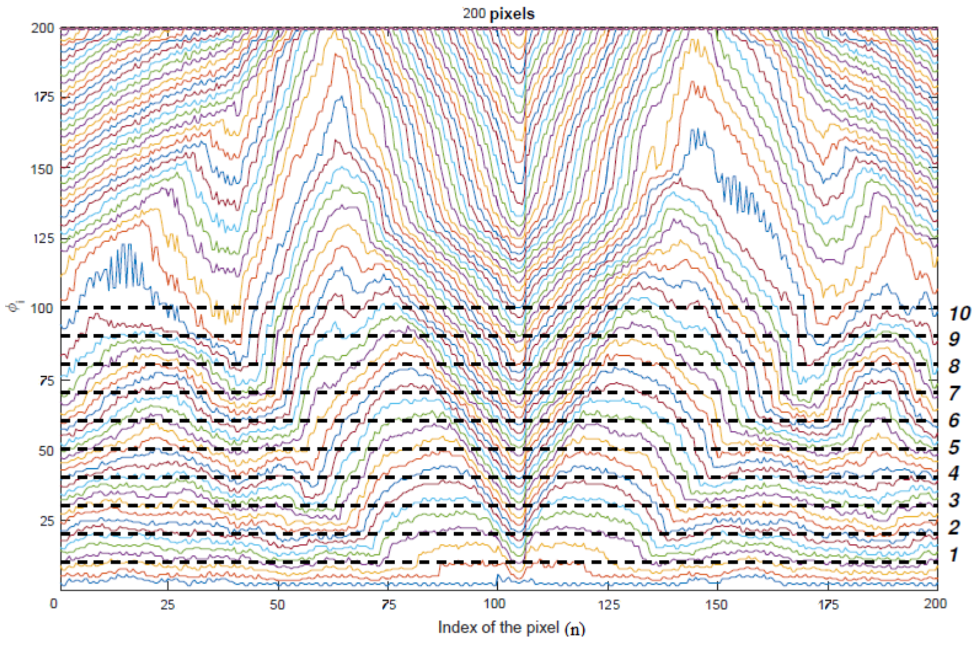


Figure 4.17: The proposed dynamic range of the increments δ with respect to the number of connected elements.

Table 4.2: Minimum number of observation at $t_{\circ}$.

Dataset	n	accuracy	time (second)
d4	55	99.56	7.47
d5	45	100	2.84
d6	35	98.98	4.56
d7	65	95.83	6.13

the accuracy scores of $30 \leq n \leq 200$ using $T = 1, 2, ..., 10$ levels are used as a cost function. Note that the levels of t are equivalent to $5\%, 10\%, ..., 50\%$ of n respectively as shown in Figure. 4.17. We found that T, for most of the samples using varies value of n, lies in this range. In general, the system performance depends on the number of observation as shown in Figure. 4.18. For each (n), we found the maximum accuracy of the different threshold values.

We also explore the optimal value of n that provides the highest discriminative representation between samples when $\delta = 0$. This is implemented using TOPSIS by searching the optimal n at $t_{\circ}$. Table 4.2 shows that the range between 35-65 pixels is the efficient number to represent the shapes among all the datasets using only $t_{\circ}$.

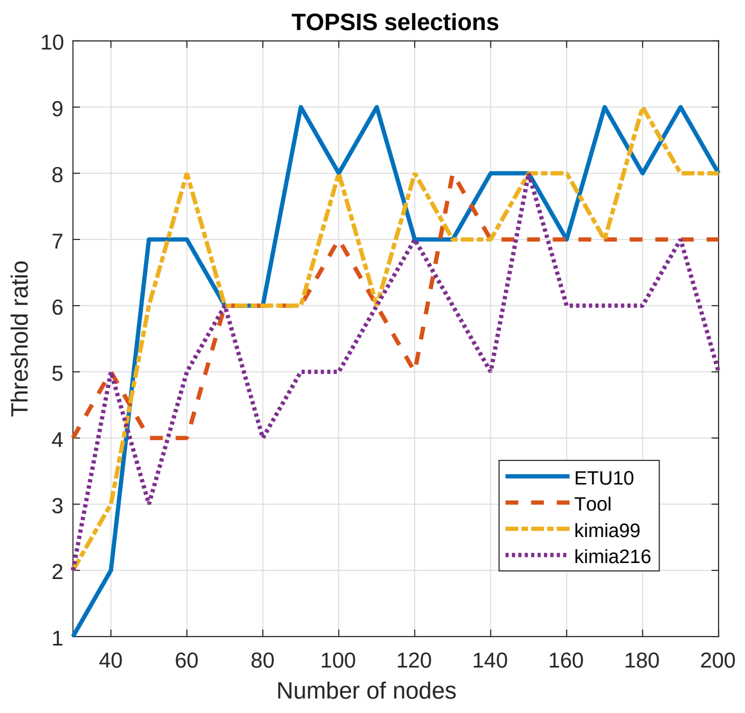


Figure 4.18: The proposed dynamic range of the increments δ with respect to the number of connected elements.

4.3.5 Time requirement

The average time to perform graph generation, features extraction, training and testing samples twelve times is also shown in Table 4.2 for all the datasets. In addition, the average required time to test a new sample was 12 milliseconds, which reflects the real-time performance of the proposed method.

4.3.6 Comparison with other works

2D shape

In order to test the robustness of the proposed method, we compare the graph spectral features (GSF) with the state-of-the-art studies. Table 4.3 shows the recognition score of the proposed method and the maximum score of the existing works for the four datasets. Note that, the proposed method performance is compared with the best available results based on the classification score and we exclude the retrieval results. The proposed method performs better than the existing methods using the ETU10 silhouette, and Tool dataset. Although there is a significant

Table 4.3: Comparison with the state-of-the-art studies (%).

Dataset	**Proposed method**	Existing work
d4	99.58	97.50 [143]
d5	100	90.86 [75]
d6	100	100 [75]
d7	96.3	98.15 [16]

Table 4.4: Classifiaction accuracy results of SHERC2010 dataset and 3D shape benchmark (%).

d8 (SHREC2010)	Average accuracy (%)	**d9** (3D benchmark)	Average accuracy (%)
f1-features [150]	86.49	3D shape histogram [87]	43.42
GPS-embedding [151]	88.87	Shape distribution [152]	67.37
Shape-DNA [153]	90.96	Complex-network [92]	70.79
Proposed method	**93**	**Proposed method**	**76.32**

similarity in these datasets, the proposed method detects these local details with a high level of detail.

3D shape

In order to demonstrate the robustness of 3D shape recognition, we test GSF against various state-of-the-art methods, whose respective configurations align with the proposed method (*i.e.*, threshold set, features, and node distribution). Table 4.4 shows that GSF exceeds the state-of-the-art performance by 2% for the **d8** and 6% for the **d9**.

4.4 Concluding remarks

In this chapter, a new method for 2D/3D shape recognition has been proposed. An adaptive graph for individual shape has been generated to fit the geometric details using the proposed threshold. Then, a combination of graph spectral and node domain features have been used, which were able to capture the global and local details of the shape. They were also invariant to rotation and scaling changes. A cross-validation technique has been applied to train and test all samples in the datasets. The proposed features have been classified using K-nearest neighbour classifier. Four public and well-known datasets were used to evaluate 2D models: the ETU dataset, Tool dataset, kimia99, and kimia216 datasets and two public, challenging datasets were used to evaluate 3D models: SHREC2010 and 3D benchmark dataset. The performance evaluation shows that the proposed method is robust in detecting different kinds of complex shapes and outperformance the state-of-the-art studies by increment 2%, 9%, 2%, and 6% for four datasets. In the next chapter, we will introduce a new method for simplifying the structure for matching purposes, which is applied for 2D/3D shape recognition and hand gesture recognition.

Chapter 5

Shape simplifying on graph spectral domain for shape recognition.

5.1 Introduction

In the previous two chapters, we solved the problem of matching low and medium-level structures. In this chapter, we will explore the graph spectral features matching for the high-level structures. Graph Signal Processing provides an appropriate mathematical platform for understanding and analysing the unstructured data, often referred to as non-uniform grids. This is because the normal Discrete Signal Processing (DSP) deals only with structured data. Graphs contain nodes connected by edges and these edges take the form of any relationship between nodes. Based on the graph connectivity, the graph spectral domain provides a rich knowledge about the structure characteristic.

A considerable amount of literature has been published on using graphs for shape matching [23–27, 31, 32]. These studies usually generated graphs based on the shape abstract to simulate the structure with low complexity details. Then, either approximate or bipartite matching methods are used for identification. However, most of these studies rely on the global details for matching. Also, the time requirements are usually too expensive for implementation. The available datasets of the 2D and 3D shape matching provide more complex shapes, making the classification process a difficult task to be implemented. Therefore, more details about the shape are needed to have a deep understanding of the structures such as local details, global details, shape simplifying. In addition, the available classes usually come with different orientation and scale, which adds an extra challenge for the classification process.

In this chapter, we are interested in using graph spectral domain features for hand gesture recognition, 2D binary shape recognition, and 3D point cloud shape recognition. The main reasons for using graphs are: 1) The graph spectral bases depend on the relative measurements between the nodes, making them invariant to the rotation changes. 2) The ability to create a shape's map with low complexity. 3) Graph connectivity observes the details of the global structure and characterises it in the graph spectral bases. The proposed method contains two parts: firstly we extract the contour and a fully connected graph is created over the contour representation to describe the global shape details using graph spectral domain features. Secondly, we extract the skeleton representation to capture the local details and to simplify the shape for matching purposes [154]. At the final step, these features are concatenated to be classified using a machine learning technique. Eight public datasets are used to evaluate the proposed method, including four hand gesture datasets, two 2D shape datasets, and two 3D shape datasets. Performance evaluation demonstrates the efficiency of the proposed method compared to the state-of-the-art studies with real-time implementation. The main contributions of this work are:

- Proposing a new method for shape recognition that preserves local and global details of shapes.
- Using the graph spectral features for hand gesture recognition.

This chapter is structured as follows, Section 5.2 illustrates the proposed method in detail including graph concepts, the silhouette and skeleton representation, and the proposed features. The performance evaluation of the proposed method using publicly available hand gestures datasets and shape datasets are presented in Section 5.3 including testing different classifiers and parameters. Section 5.4 compares between the proposed methods in this thesis. This work will be concluded in Section 5.4.

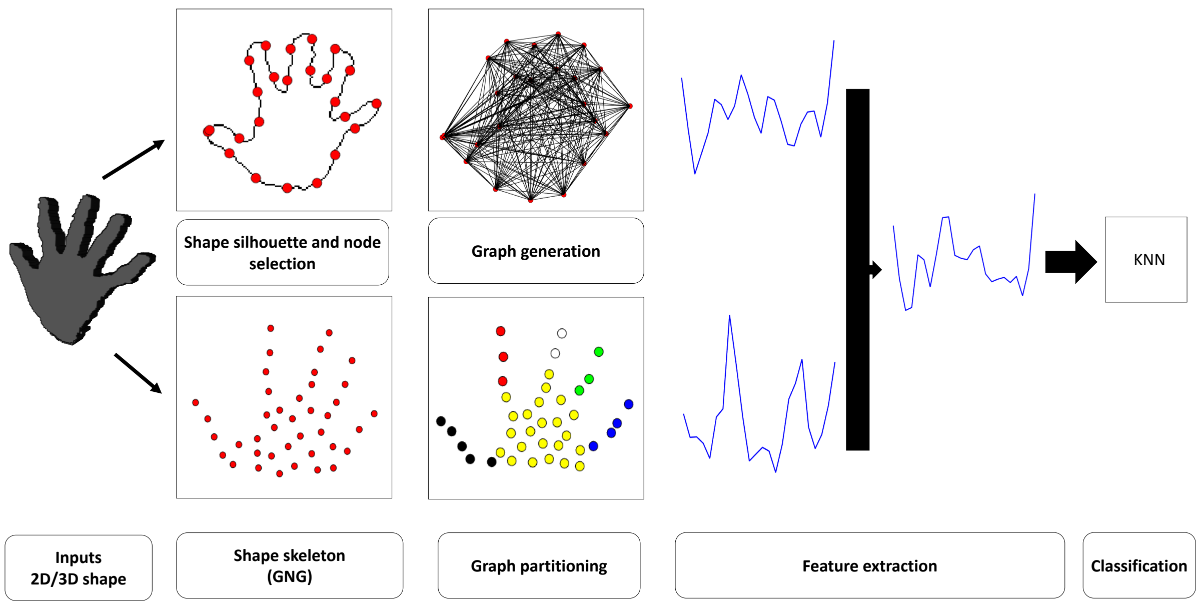


Figure 5.1: The pipeline of shape representation, which is extracted from the silhouette and skeleton representations. The top row indicates a silhouette representation and graph generation to extract the features. While the bottom row is shown how to form the shape skeleton to simplify the shape into several parts. Both features are concatenated and classified using a machine learning technique at the final step.

5.2 The proposed method

The full pipeline of the proposed method is shown in Figure. 5.1. The method starts by extracting the silhouette and skeleton representations of the shape. Candidate nodes are selected from the contour to form a fully connected graph, which is used to characterise the topology of the shape. A skeleton representation is employed to partition the shape into meaningful parts, which helps understanding the local details. At the last step, a combination of silhouette and skeleton features are used to classify shapes using a machine learning technique. Details of each step are provided in following subsections. Before we begin to explain the proposed method, relevant graph properties are provided in the next subsection.

5.2.1 Graph preliminaries

For a given graph $\mathcal{G}$, we calculate the adjacency matrix, the combinatorial graph Laplacian matrix, normalised graph Laplacian matrix, degree matrix, geometric graph Laplacian as shown

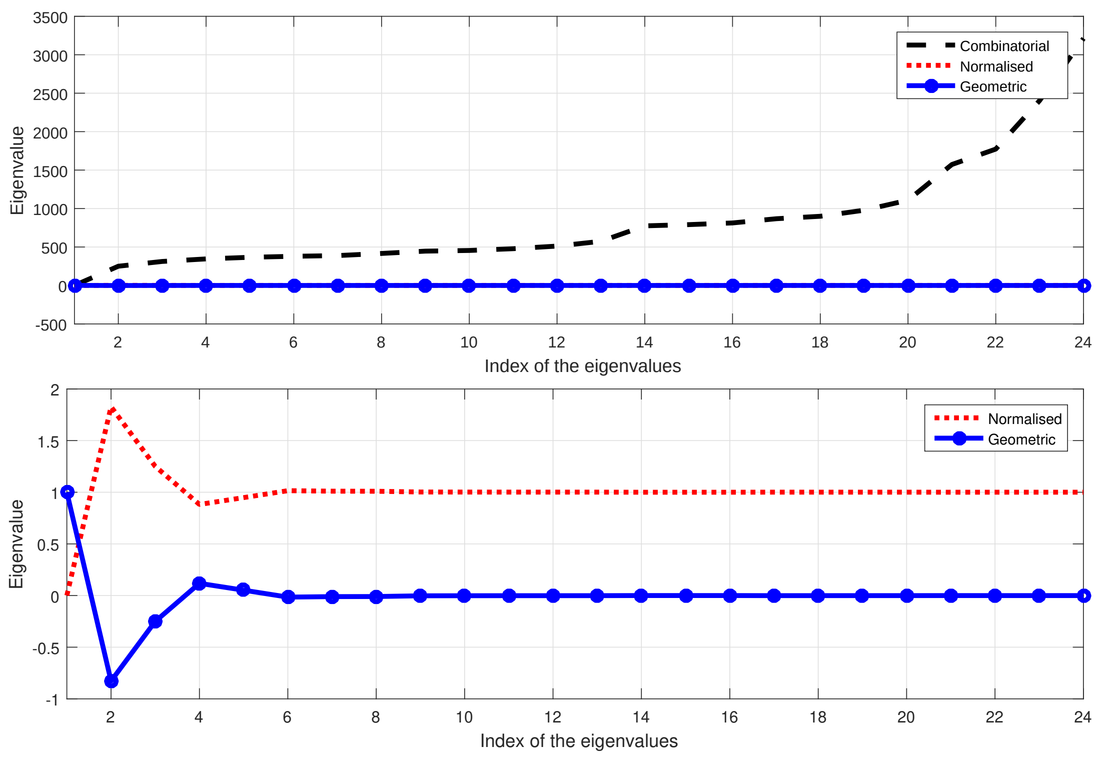


Figure 5.2: (Top): The eigenvalues of the L,$\mathcal{L}$ and Γ. (Bottom): zoom in between the $\mathcal{L}$ and Γ eigenvalues.

in Eq. (2.1), Eq. (2.3), Eq. (2.4), Eq. (2.5), Eq. (2.6), and Eq. (2.7) respectively.

The eigenvalues of $\mathcal{L}$ have different behaviour from the combinatorial Laplacain version, where the $\lambda_0 = 0$ and $\frac{1}{n}\sum_{i=0}^{n-1}\lambda_i \approx 1$. In other words, the eigenvalues are fluctuated around 1 and they are ordered as $0 = \lambda_0 < |\lambda_1 - 1| \geq |\lambda_2 - 1| \geq ... \geq |\lambda_{(n-1)} - 1|$. The eigenvalues of Γ have almost the same behaviour of $\mathcal{L}$, but are ordered as $|\lambda_0| \geq |\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_{(n-1)}|$ except that they are fluctuated around the zero. Figure. 5.2 shows the general trends of these eigenvalues for undirected fully connected graph. Note that, these bases could be changed with the size of the graph or the connectivity as shown in **Chapter 4**. More information about the graph connection can be found in Section 4.2.2.

Since we deal with different shapes (*i.e.*, different graph structures), a combination of global and local details is required to classify shapes. Therefore, in this work, we use a silhouette with a fully connected graph to observe the global detail. In addition, we use the skeleton representation to capture local details by simplifying the structure into several parts through the partitioning process. In the next subsections, we will provide information about the shape representations.

5.2.2 Silhouette and Skeleton representation

The silhouette and skeleton representation are similar to the procedure in Section 4.2. We implement an edge detector (*e.g.*, Sobel filter) of P with length (N) in order to reduce the complexity and to keep samples in the same size. The resulting 2D closed path consists of random pixels, which are used to generate a new set of pixels (n_1) to form a new down-sampled path, $\hat{P}$, as shown in Eq. (3.1) (as in the top sub-figure of Figure. 5.1). We also used GNG to generate a new set of nodes (n_2) inside the shape as shown in Figure. 5.1.

5.2.3 Silhouette-based graph spectral feature extraction

In order to classify different shapes, we have to understand the boundary's map of the shape. This study therefore interprets the contour details through the graph spectral domain. We take advantage of the eigenvectors and eigenvalues representations of the fully connectivity to recognise shapes. Before we explain the proposed features, we will give insight about how these spectral bases preserve the topology of the shape and how the eigenvalues can be used for shape matching.

There is a vast literature discussing the relation between graph bases and connectivity. For example, for any node $\mathcal{V}_i$, the Fiedler value is inversely proportional to the weight of the edge connecting i and j in the adjacency matrix [35,43,44]. According to this concept; dense areas or a group of nodes, which are close together, are presented as high values compared to other values in the graph bases. This can be clearly seen in the extended fingers for hand gesture recognition and limbs or other small parts in shapes. Figure. 5.3 illustrates these concepts using 6 classes of hand gestures and we can see that the peaks on the graph spectral basis refer to the extended fingers. The sequence of the peaks and troughs helps to localise the extended fingers with respect to the full contour details. This leads to classify shapes, which have the same number of extended parts.

In the next experiment, we test individual graph eigenvector of graph Laplacian matrix to determine the most effective basis. As can be seen in Figure. 5.4, a wide range of accuracy

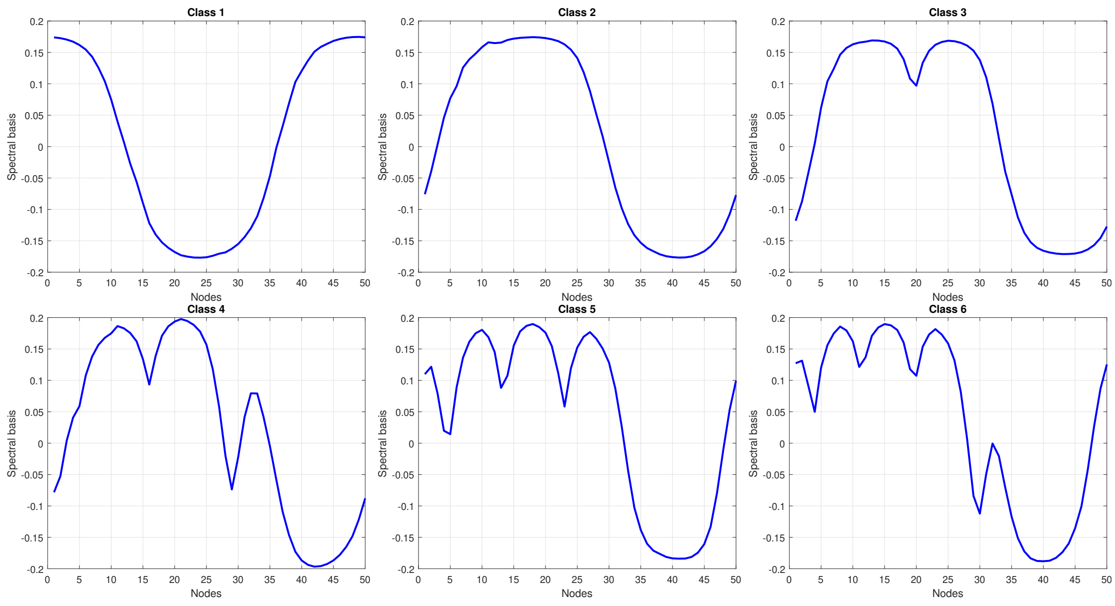


Figure 5.3: Graph spectral response (*e.g.*, the first eigenvector of the normalised Laplacian matrix) of different classes using dataset (**d10**). From left to right, top row (classes 1-3) and the bottom row (classes 4-6) .

levels is achieved. The bases can be categorised into four sections (A, B, C, D). Both normalised and geometric graph Laplacian matrix recognise shapes with a high level of accuracy at (A-section). Then, it drops towards (D-section). Whereas the combinatorial version provides inverse behaviour to the normalised the geometric versions and it reaches the highest accuracy at (D-section). This experiment shows that the A and D sections of the eigenvectors provide a significant contribution of the matching percentage, whereas the rest of the eigenvectors (*i.e.*, B-section and C-section) show lower range of accuracy. Therefore, this work employs a certain range of the bases to eliminate the effect of the inefficient eigenvectors.

Since the graph eigenvectors are presented as a scaler values from [-1,1] to reflect the relative measurements of the nodes connectivity, it is not affected by the linear changes. Therefore, the graph eigenvalue is used to provide a notation of the structure size. For illustration, an experiment has been conducted to test the graph bases for the same shape in its original size with a scaled size as shown in the top of Figure. 5.5. These two pictures have exactly the same eigenvectors but different eigenvalues. It also shows the difference between these eigenvalues in the bottom. The large difference between these shapes occurs in the last eigenvalue. Therefore, the last λ_{n_1-1} and second eigenvalue λ_1 are used as features.

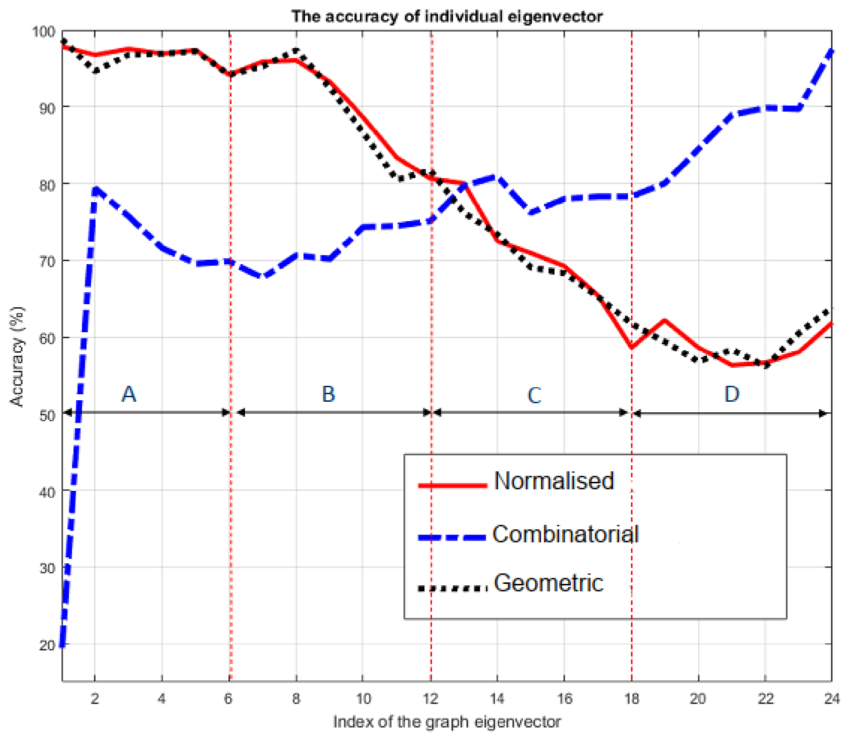


Figure 5.4: The accuracy score for individual graph eigenvectors. In this experiments, we use different graph eigenvectors of the three different Laplacian versions to compute the accuracy level using KNN classifier.

Assume that S_Q is a matrix, which consist of $(n/4)$ eigenvectors in Q-section as shown in Eq. (5.1). We calculate the mean value of each node in the eigenvectors of Q-section so that the average values have the same length as the eigenvectors as shown in Eq. (5.2).

$$S_Q = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ . \\ . \\ \mathbf{u}_{((N/4)-1)} \end{bmatrix} \tag{5.1}$$

$$|S_i| = \frac{1}{(N/4)} \sum_{j=0}^{(N/4)-1} \mathbf{u}_{(i,j)}, \qquad i = 0, 1, .., N-1, \tag{5.2}$$

where $||$ is a function to find the mean, and j is the index of each eigenvector. Figure. 5.6 shows an example of how to compute the mean value of eigenvectors. This is implemented to achieve

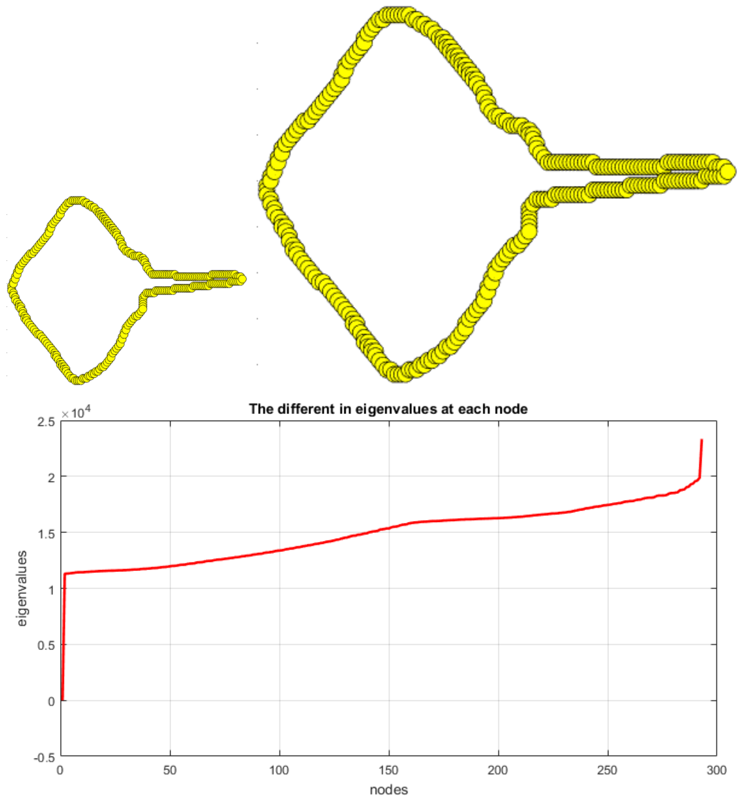


Figure 5.5: In the top, the original size of the shape(Left) and its scaled version 2X (Right). In the bottom, the different in eigenvalues at each node between two samples above.

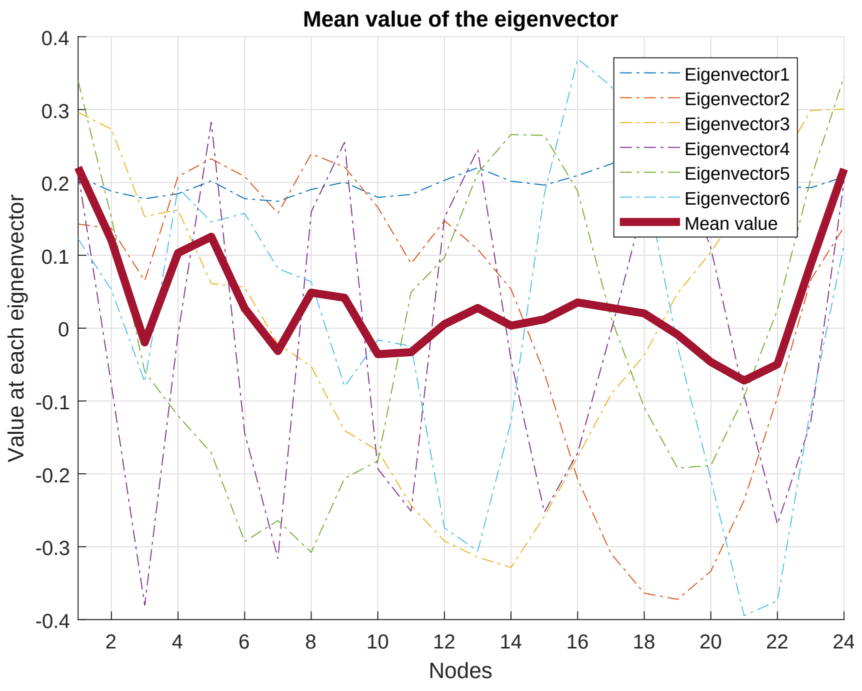


Figure 5.6: Mean value of the eigenvectors in A-section. This experiment demonstrates how to calculate $|S_A|$, which represents the mean of the six eigenvectors in section A. The result is a vector of n length.

a stable representation of the shape. Also, using the mean value reduces the complexity of the system instead of taking all the eigenvectors. The general description of the proposed features based on the silhouette representation is obtained as in Eq. (5.3).

$$F = e^{(\lambda_{(1)} - \lambda_{(n-1)})}|S|, \tag{5.3}$$

In order to optimise these features, we consider the accuracy state in Figure. 5.4. Therefore, the optimal features are computed as shown in Eq. (5.4), Eq. (5.5), and Eq. (5.6) for the combinatorial, normalised, and the geometric versions respectively.

$$F_L = e^{(\lambda_{(1)} - \lambda_{(n-1)}/1000)}|S_D|, \tag{5.4}$$

$$F_{\mathcal{L}} = e^{(\lambda_{(1)} - \lambda_{(n-1)})}|S_A|, \tag{5.5}$$

$$F_{\Gamma} = e^{(\lambda_{(1)} - \lambda_{(n-1)})}|S_A|. \tag{5.6}$$

The silhouette feature length = n_1.

5.2.4 Skeleton based partitioning and graph features

In this subsection, we provide a fast partitioning method, which can be used to simplify the structures for matching purposes. Graph partition has a long history in the segmentation process specifically using the eigenvector corresponding to the smallest non-zero eigenvalue, which is known as a Fiedler vector. This is because Fiedler vector provides the minimum cutting ratio according to the optimisation formulae Eq. (2.13) [42]. The vast majority of current partitioning methods require determining the number of clusters in advance. For this purpose, we adapt new rules to achieve a fully automatic and stable recursive hierarchical partitioning, which leads to automatically identifying the meaningful parts of the structure. That is to say, there is no need for human intention to determine the number of required clusters.

For graph partitioning, the main differences in terms of graph generation is that we use the combinatorial Laplacian version and conditional connectivity, where the condition here is the smallest distance to link all nodes as one group. The main reason for using a combination of the conditional connectivity and the combinatorial Laplacian matrix is that the local details of the shapes can be efficiently detected as shown in **Chapter 4**. Thus, limb nodes always have less connected nodes compared to the other nodes in the main area, and that makes the segmentation process an easy task to be implemented. Repeating the procedure will end up with efficient segmentation quality. In order to avoid fragmentation in the main body and in the small parts, the segmentation process should satisfy two rules, which are:

1. The minimum number of nodes in each sub-group = 2.

$$n_{g_1} \geq 2 \quad \text{and} \quad n_{g_2} \geq 2. \tag{5.7}$$

2. In order to split two graphs (g_1, g_2), the difference in number of nodes between them $> n_2/3$, where n_2 the total number of nodes in the skeleton representation.

$$|n_{g_1} - n_{g_2}| > n_2/3, \tag{5.8}$$

where n_{g_1} and n_{g_2} are the number of nodes in the new generated sub-graphs. Figure. 5.7 shows nine levels of 3D shape segmentation as an example. For each given object, we calculate its features $\text{GSF}_k = \{f_a, f_b, f_c\}$ as follow:

1. Number of clusters

$$f_a = \omega. \tag{5.9}$$

2. Fiedler value at each node by the end of segmentation

$$f_b = \hat{\mathbf{r}}_i \, \mathbf{u}_{(1,i)}, \qquad\qquad i = 0, \ldots, n-1. \tag{5.10}$$

3. We also consider the number of connected nodes at each node (*i.e.*, node degree)

$$f_c = \hat{\boldsymbol{\Phi}}_{\boldsymbol{i}}, \qquad\qquad i = 0, \ldots, n-1. \tag{5.11}$$

The skeleton feature length = $2n_2 + 1$.
The total feature length = $n_1 + 2n_2 + 1$.

5.2.5 Machine learning

Several experiments have been implemented to test the optimal classifier for classification as will be shown in the next section. The test includes K-Nearest Neighbour classifier with K=1 (KNN), multi class Support Vector Machine with Cubic kernel (CSVM), Classification Tree with 30 learners using the bag-ensemble method (CT), Discriminative Analysis with pseudo Quadratic as a discriminative type (QDA), and Neural Network with two layers feed-forward network and 30 neurons in its hidden layer (NN).

5.3 Performance evaluation

In this section, we provide information about the implementation of the proposed method including, the datasets used for evaluation, different classifiers test, confusion matrices, time re-

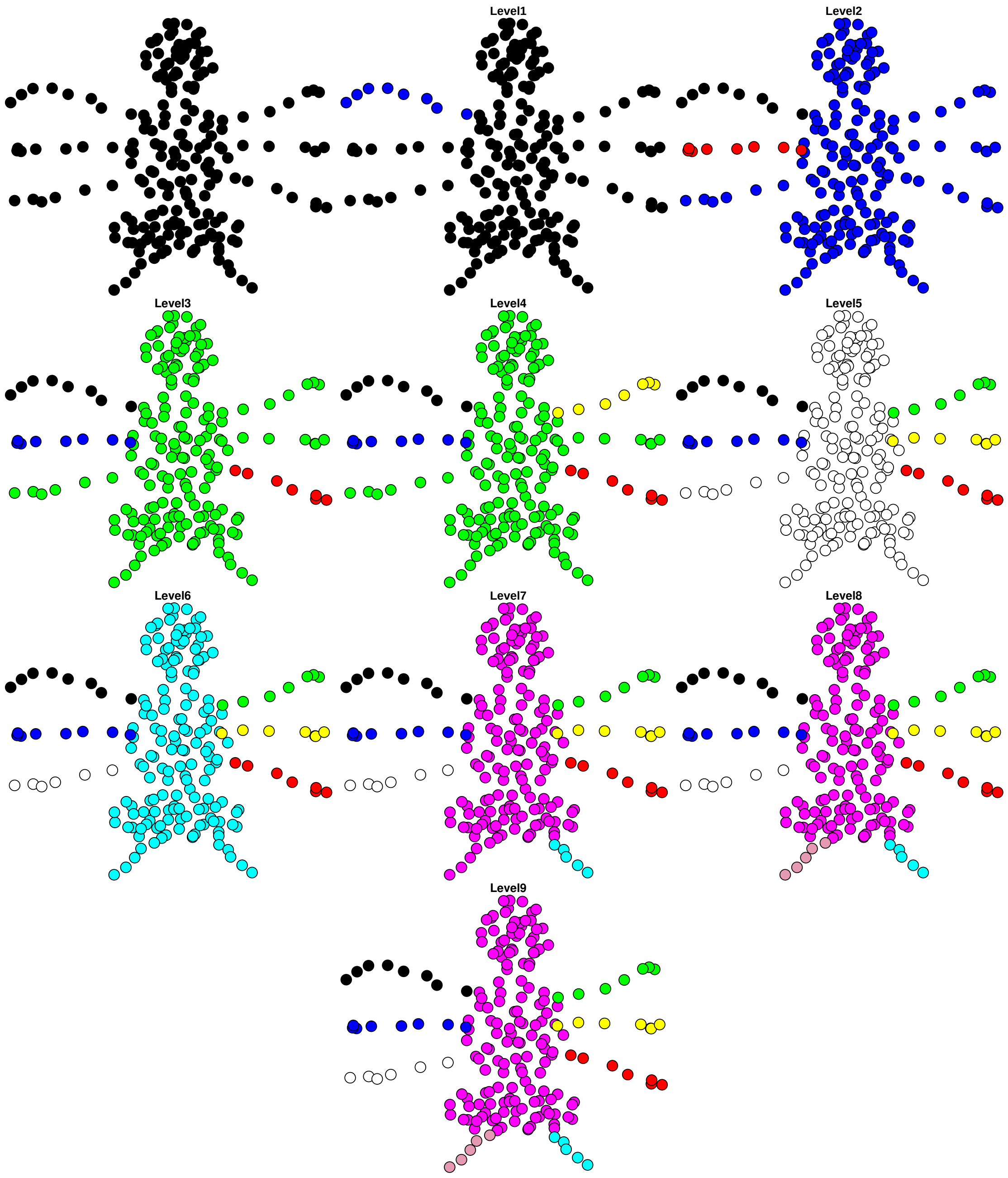


Figure 5.7: Segmentation procedure of the proposed graph partitioning. In this example, we show nine segmentation levels, and as we can see that there is no different between level right and level nine and therefore the segmentation process stop. More samples can be found in the Appendix C.

quirement, and comparison with other studies. All the experiments were implemented using MATLAB R2018a on a PC with Intel processor, CPU@3.6GHz and RAM 16GB. A 10-fold cross validation scheme is used to train and test most of the datasets. In order to reduce the complexity, a few number of nodes are used to generate the silhouette $(n_1 = 50)$ for 2D shapes and the skeleton $(n_2 = 200)$ for 3D shapes.

5.3.1 Datasets

A big number of data is used to evaluate the proposed graph spectral feature including variety forms of shapes starting from simple shapes to more complex details such as: 2D RGB images **d10**, depth images (**d11**, **d12**, **d13**), 2D binary shapes (**d4**, **d6**), and 3D point cloud samples (**d8**, **d9**). More details about these datasets are shown in the next subsection.

1. **d10**

   The first dataset (RGB dataset) [155] contains 36 American Sign Language (ASL) gestures implemented 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 gesture as shown in Figure. 5.8.

2. **d11**

   The second dataset (RGB-depth dataset) [99] contains 10 subjects $\times$ 10 hand gestures $\times$ 10 different orientations = 1000 colour and its corresponding depth images as can be seen in Figure. 5.8. The dataset includes the subject poses with various hand orientation, scale and articulation. Only depth images are used for evaluation.

3. **d12**

   The third dataset (Synthetic dataset) [102] 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 Figure. 5.8.

4. **d13**

   The fourth dataset (HKU dataset) [116] 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 for evaluation as shown in Figure. 5.8.

5. **d4**, **d6**, **d8**, **d9**

   Four datasets are used as well for evaluation, which are two datasets for 2D shapes (**d4**, **d6**) and another two datasets for 3D shape (**d8**, **d9**). More details about these datasets can be seen Section 4.3.

5.3.2 Cross-validation of different classifiers and Laplacian matrices.

In this experiment, we apply L for shape partitioning. In order to observe the global details, we examine the three versions of the graph Laplacian matrices based on different classifiers as can be seen in Table 5.1. Three points can be concluded from these results, which are firstly KNN, CSVM and NN provide the highest recognition score and we use KNN in the rest of the experiments because of its fast implementation. Secondly, Γ and $\mathcal{L}$ show approximately the same performance, and we use Γ for the rest of the experiments in this chapter. Finally, the correlation of the results based on different classifiers demonstrate the distributions of the features among different classes.

5.3.3 Confusion matrices

In this study, we use the confusion matrix to show the performance of the proposed method and the classification results of the individual dataset. The confusion matrices of the datasets are shown in Figure. 5.9. However, we exclude **d5** and **d6** because the accuracy score of these datasets is nearly 100%. The results demonstrate the efficiency of the proposed method for shape classification by achieving 99.53%, 99.6%, 95%, 98.5%, 99.6%, 100%, 95%, and 78.68% for **d10**, **d11**, **d12**, **d13**, **d4**, **d6**, **d8**, **d9** respectively.

In general, there are no serious confusion cases for all datasets and the only challenge is in case when the classes have almost the same contour structure as can be seen in Figure. 5.10. The fist and the open hand have nearly the same contour properties, which result in the same

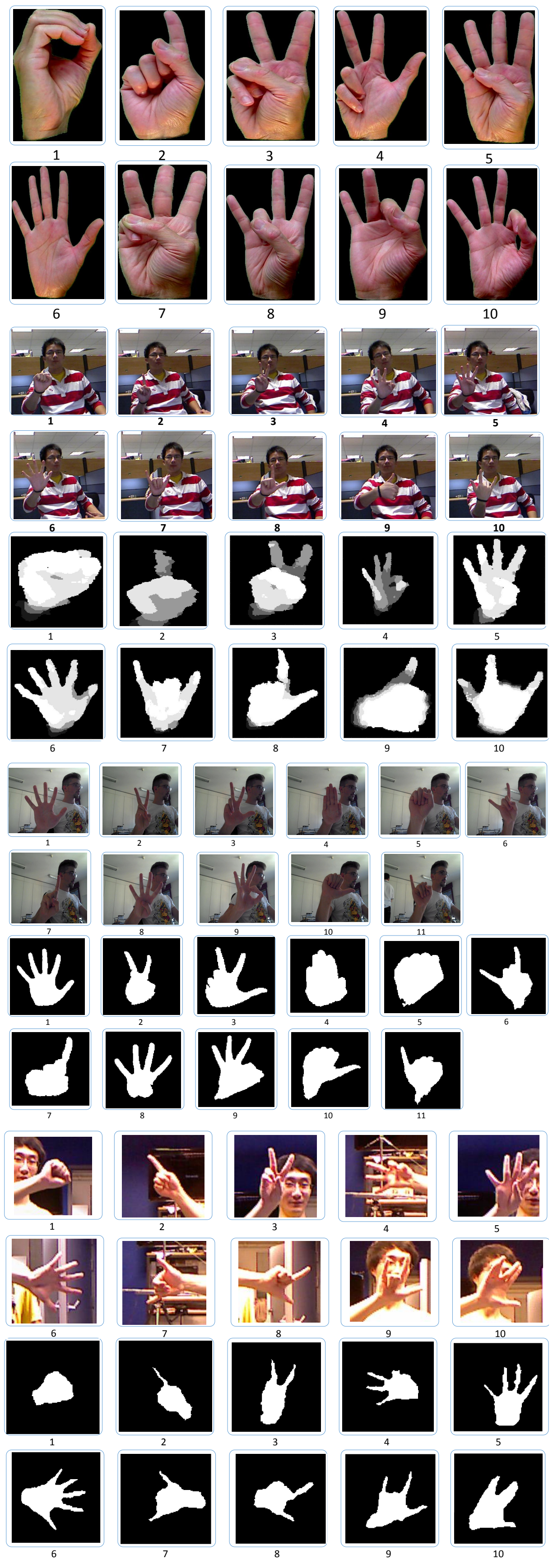


Figure 5.8: Samples of the datasets (**d10**, **d11**, **d12** and **d13** from the top to the bottom) were used to evaluate the proposed method.

Table 5.1: The accuracy score (%) of different classifiers for the three graph Laplacian matrices.

	KNN	CSVM	CT	DA	NN
d10					
L	93.2	98.7	91.1	97.2	98.6
$\mathcal{L}$	98.7	98.6	90.3	98.9	99.1
Γ	99.53	99.1	91.3	98.8	**99.67**
d11					
L	94.2	97.19	91.34	96.38	95.1
$\mathcal{L}$	99.6	98.84	92.42	98.19	98
Γ	**99.7**	98.27	88.11	97.57	99.1
d12					
L	87.48	91.66	79.60	84.92	89.7
$\mathcal{L}$	94.4	93.03	76.19	91.28	91.4
Γ	**95**	92.44	71.81	90.15	91.3
d13					
L	93.1	94.7	84	89.3	93.9
$\mathcal{L}$	98.3	93.4	72.7	91.4	91
Γ	99.4	**95**	78.1	95.1	94.2
d4					
L	94.3	94.7	86.8	93.61	96.2
$\mathcal{L}$	98.3	97.6	84.7	96.12	99.5
Γ	99.6	98.8	85.2	96.43	**99.7**
d6					
L	89.90	92.93	88.89	92.93	93.8
$\mathcal{L}$	100	97.98	85.86	96.97	99
Γ	**100**	97.98	85.86	97.98	99.6
d8					
L	88.63	89.2	75.8	87.82	91.1
$\mathcal{L}$	**95.47**	94.8	70.1	91.64	93.8
Γ	95	94.5	71.4	93.47	95.2
d9					
L	66.7	65.4	65.47	62.39	68.25
$\mathcal{L}$	76.23	72.9	62.61	70.46	75.4
Γ	**78.68**	77.3	64.32	74.12	78.3

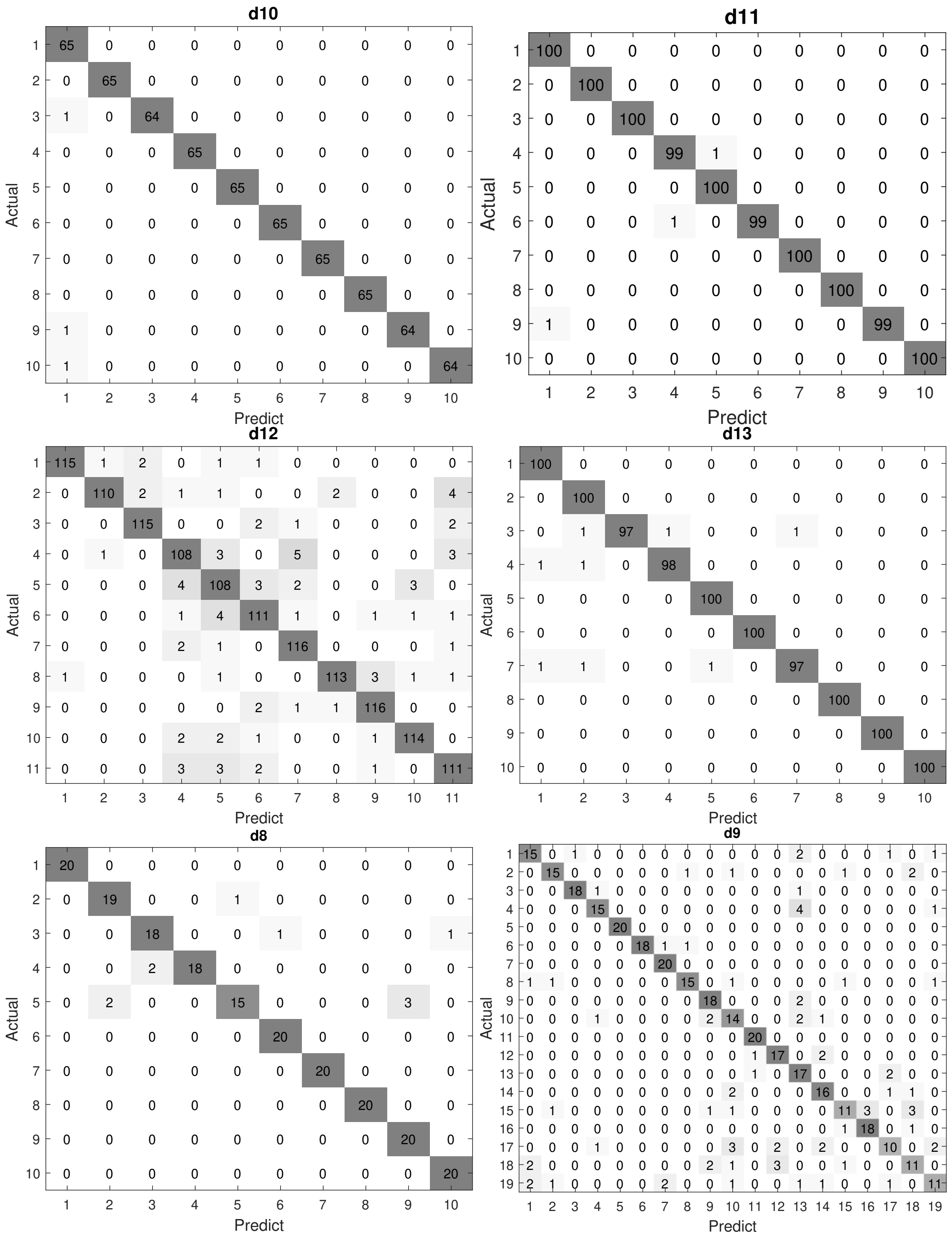


Figure 5.9: Confusion matrices using KNN classifier.

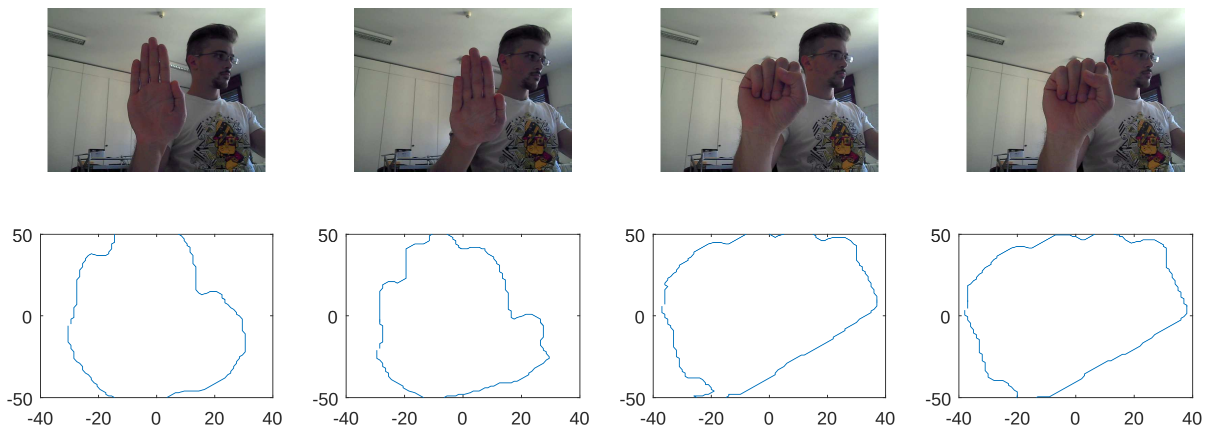


Figure 5.10: Confusion cases, where the silhouette representations of different classes are similar.

Table 5.2: The average time to perform different steps of the proposed method.

Step	Performance average time (ms)
GNG	3984.54
Partitioning	428.12
Graph generation (silhouette)	0.429
Feature extraction	0.185
Classification	1.654
Full time system	$\approx$ 4.4 seconds

segmentation process. These classes are detected based on the silhouette representation rather than skeleton representation.

5.3.4 Performance complexity

Table 5.2 shows details implementation time for each individual step using different datasets. The total required time is only above four seconds and can be improved with other alternative programs. Except for GNG training time, all other actions are in real time. Note that, the training time of the GNG depends on several parameters such as the required number of nodes, number of iterations and other parameters.

Table 5.3: Comparison with the existing works using **d10**.

d10	Method	Score (%)
Proposed method	GSF	**99.53**
Aowal *et al.* [110]	DZM	98.51
Otiniano *et al.* [111]	ZM	96.37
Ng and Ranganath [113]	FD	95.68
Avraam [108]	GA	83.1

Table 5.4: Comparison with the existing works using **d11**.

d11	Method	Score (%)
Feng *et al.* [115]	GSF	100
Proposed method	GSF	**99.7**
Wang *et al.* [116]	SP-EMD	99.6
Zhang *et al.* [109]	H3DF	95.5
Ren *et al.* [99]	FEMD	93.9
Plouffe and Cretu [98]	DTW	93.88
Chen *et al.* [117]	FS	93.2
Zhang *et al.* [109]	HOG	93.1

Table 5.5: Comparison with the existing works using **d12**.

d12	Method	Score (%)
Proposed method	GSF	**95.00**
Minto and Zanuttigh [102]	LCDT	89.91

5.3.5 State-of-the-art performance

In order to test the robustness of the proposed graph spectral features, we compare the method with the state-of-the-art studies. Table 5.3, Table 5.4, Table 5.5, Table 5.6, Table 5.7, Table 5.8, Table 5.9 and Table 5.10 show the recognition score (%) of the proposed method and the existing works for all the datasets using different methods and classification techniques. The proposed method exceeds the state-of-the-art performance using **d10**, **d12**, **d4**, and **d9** by 1.02%, 5.09%, 2.1% and 7.89% respectively.

Table 5.6: Comparison with the existing works using **d13**.

d13	Method	Score (%)
Proposed method	GSF	**99.4**
Wang *et al.* [116]	SP-EMD	99.1

Table 5.7: Comparison with the existing works using **d4**.

d4	Method	Score (%)
Proposed method	GSF	**99.6**
Akimaliev and Demirci [143]	Shape abstraction	97.5
Demirci *et al.* [66]	Skeleton	95.2

Table 5.8: Comparison with the existing works using **d6**.

d6	Method	Score (%)
Proposed method	GSF	**100**
Nanni *et al.* [75]	Local descriptors	100

Table 5.9: Comparison with the existing works using **d8**.

d8	Method	Score (%)
Agathos *et al.* [156]	Graph matching	97.6
Proposed method	GSF	**95**
Lavoue [157]	Hybrid 2D/3D	92.5
Lian *et al.* [90]	CboFHKS	90.1
Reuter *et al.* [153]	Shape-DNA	90.96
Chaudhari *et al.* [151]	GPS-embedding	88.87
Khabou *et al.* [150]	f1-features	86.49

Table 5.10: Comparison with the existing works using **d9**.

d9	Method	Score (%)
Proposed method	GSF	**78.68**
da Silva and Backes [92]	Complex network	70.79
Osada *et al.* [152]	Shape distribution	67.37
Ankerst *et al.* [87]	3D shape histogram	43.42

5.4 Concluding remarks

The new emerging field of graph spectral domain provides new opportunities to exploit the geometric structure of the complex shapes. This work has presented a new method for 2D/3D shape matching and hand gesture recognition based on the shape silhouette and skeleton representation. A fully automatic divisive hierarchical clustering method is used to simplify the shape and capture its local details. A set of spectral graph features is classified using a KNN classifier. Eight public datasets were used for evaluation including hand gesture recognition, 2D binary shape and 3D shape point cloud. The proposed method has shown an improvement compared to the state-of-the-art methods for four datasets by 1.02%, 5.09%, 2.1% and 7.89%.

Chapter 6

Conclusions

6.1 Summary of achievements

This thesis have explored the utility of the graph spectral domain features for shape recognition. Different graph formulations have been presented that observe topological details. A new dataset for in-air hand-drawn number and shape recognition have been introduced. We have also taken into account different shape characteristics, including in-air hand-drawn recognition, hand gesture recognition, and 2D/3D shapes recognition. Three different methods have been proposed, each based on global details, local details, and a combination of both respectively.

The first problem was to capture the overall structure of in-air hand-drawn shapes. We have presented novel graph spectral features for in-air hand-drawn number and shape recognition. The proposed method includes pre-processing for converting the hand path movement, captured via Kinect, into a fully connected graph followed by analysis of the eigenvectors of the normalised Laplacian of the graph adjacency matrix to form the feature vector. We utilised the eigenvector, $\mathbf{u}_0$, as this captures the topology details of the graph. The proposed method has resulted in the highest performance with accuracies of 99.56% and 99.44%, for numbers and shapes, respectively, out performing existing methods for three different datasets. The proposed method also has the added benefits of fast operation and invariance to rotation and flipping.

The second problem required developing a new optimal model for shape recognition. The proposed model has the ability to observe fine details of the shape and characterise the topology of the shape using an adaptive graph connectivity. Then, a set of graph spectral features was used to capture the global and local details of the shape. The performance evaluation showed that the proposed method is robust in detecting different complex shapes with improving results compared to the state-of-the-art studies of the ETU10 dataset, Tool dataset, SHREC2010 dataset, and 3D shape benchmark dataset by 2%, 9%, 2% and 6% respectively.

The third problem focuses on simplifying the shapes for matching purposes. A new method for 2D/3D shape matching and hand gesture recognition based on shape silhouette and skeleton representations has been presented. A fully automatic divisive hierarchical clustering method was used to simplify shapes and capture their local details. This is followed by feature extraction process. The shape simplification features improves the state-of-the-art performance for four datasets including RGB hand gestures, synthetic, SHREC2010, and 3D shape benchmark datasets by 1.02%, 5.09%, 2.1% and 7.89%, respectively.

Finally, we would like to highlight a few points regarding to the technical methods used in this thesis. A rotational invariant method for in-air hand-drawn shape recognition is presented in **Chapter 3**, and we showed how the number four is detecting from different directions. However, some applications require determining the directions of the hand movements. For example, it is necessary for other applications to distinguish between right and left swipes. The method in **Chapter 3** does not apply for such applications. Although it can be achieved by modifying the feature representation, then we would lose the constant rotation features. Also, in **Chapter 5**, a shape simplification method is presented. However, in this method, it is difficult to identify shapes that have the same silhouette representations. In the next sub-section, we provide a cross-validation performance of different methods.

6.2 Cross-validation of different methods

In this section, we show the cross-validation performance of the three methods namely, global-based, local-based, and simplicity-based using all dataset in this thesis. Table 6.1 illustrates a detailed accuracy score (%) for each method using all datasets. For simplicity, Figure. 6.1 visualises the contents of Table 6.1.

In Figure. 6.1(a), we can see that the global details work very well for in-air hand-drawn, but it fails to detect 2D/3D shapes because the high similarity in their structure. Therefore,

Table 6.1: Cross-validation of different methods using all the datasets in this thesis.

Application	Dataset	Reference	**Global**	**Local**	**Simplicity**
In-air hand-drawn shape recognition	In-air dataset1-**d1**	[54]	97.39	69.36	58.37
	In-air dataset2-**d2**	[60]	99.2	71.54	61.24
	Our dataset-**d3**	[127]	99.56	62.72	55.5
2D/3D shape recognition	ETU10 silhouette-**d4**	[143]	82.12	99.56	99.6
	Tool dataset-**d5**	[144]	14.28	100	100
	Kimia 99-**d6**	[145]	57.25	98.98	100
	Kimia 216-**d7**	[146]	36.8	95.83	96.1
	SHREC2010-**d8**	[147]	21.4	93	95
	3D shape benchmark **d9**	[148]	15.36	76.32	78.68
Hand gesture recognition	RGB dataset-**d10**	[155]	95.6	98.1	99.53
	RGB-depth dataset-**d11**	[99]	91.4	96	99.7
	Synthetic dataset-**d12**	[102]	89.2	93	95
	HKU dataset-**d13**	[116]	88.82	95.4	99.4

global based detection is not enough for these structures. Similarly, the simplicity and locally-based methods detect complex shapes, but cannot distinguish between simple structures because numbers like one and seven do not have much detail in their structure. It is very likely to detect the number one with small noise as number seven because the local-based method is very sensitive to fine details.

Figure. 6.1(b) shows that hand gestures are well-identified using the three methods with lower standard deviation compared to other applications. Here, we can note that the worst scenario occurs when using global detection to recognise 2D/3D shapes and this is understandable with the increasing complexity of the shapes in 2D/3D data sets.

Figure. 6.1(c) demonstrates that the overall performance for all methods and application. All the methods work very well for the hand gesture recognition because it can be considered a low-level structure in terms of skeleton representation and a high-level structure in terms of silhouettes representation. Table 6.2 highlights the main features and differences between the three technical chapters.

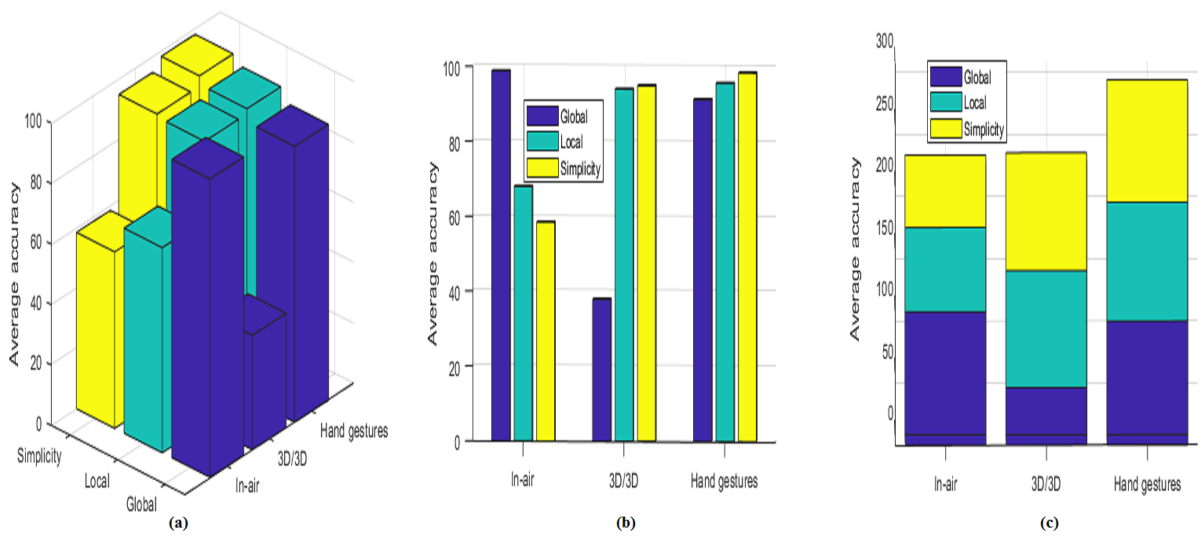


Figure 6.1: (a): Average performance for each application using a different method, (b): individual method performance for each application, (c): overall accuracy for each application.

Table 6.2: Details of the contributions.

Chapter	Aim	Laplacian matrix	Connectivity	Application
Chapter 3	Global details	$\mathcal{L}$	Fully connectivity	In-air handwritten recognition
Chapter 4	Local details	$\mathbf{L}$	Adaptive connectivity	2D/3D shape recognition
Chapter 5	Simplifying	$\mathbf{L}$ and Γ	Both	Hand gestures, 2D/3D shape recognition

6.3 Future directions

In this section, we expand the content of this thesis to some possible future directions as follows,

1. In **Chapter 3**, we proposed a robust model for in-air hand-drawn number and shape recognition. This model can be extended to detect the movements of two hands simultaneously. This requires an initial separation process to create two separate graphs or one graph. Then, the detection procedure can be performed using the spectral domain. In addition, another type of camera can be used to capture hand movement instead of a depth map. For example, an event camera that senses the movement of a scene can be a possible solution.

2. In **Chapter 4**, we demonstrated the robustness of the adaptive graph formulation for understanding shapes, based on maximum entropy. In this work, we use a statistic measurement (*i.e.*, entropy). In the future, optimisation techniques can be used to achieve optimal representation. Furthermore, deep learning technique can be also applied by training and testing the highest discriminative edges weights among the predefined dataset. In other words, for any data sets provided, we can create graphs for each sample and this is followed by training step to achieve effective representation of each sample.

3. In **Chapter 5**, we presented a new method for shape clustering based on adaptive connectivity. The proposed method was used for matching purposes. However, the separation boundaries between different cluster were approximated and not exactly on the separation line. This is because if we use GNG, we lose the homogeneous distribution of the nodes in the joining areas between two parts. Further investigation is needed in future to get better representation, so it can be proposed as an graph-based segmentation method.

4. Using adaptive graph formulations can be applied in many applications, such as image understanding. Since the generated graph highlights the most important details, resulting in efficient representation of the graph bases. Possible applications may include, Graph Convolution Neural Network (GCNN) for classification, salience map detection based on the sign of the Fiedler vector, and image compression based on the graph eigenvectors.

5. From the content of this thesis, we can conclude the importance of the graphical model for shape characteristic. This can be used to identify the human activities in general, especially if we consider each movement to be a timeline-based graphical model. Then, a bipartite graph matching can be performed to recognise different actions. In addition, a graph spectral feature can be also useful for distinguishing various tasks based on the edges between nodes.

6. In this thesis, we explore the use of the graph spectral domain features for shape representation, followed by machine learning techniques for classification step. In future, graph-based classifiers such as the deep graph regularized learning [158] could be considered to achieve a fully graphical model.

Appendices

Appendix A

In-air hand-drawn number and shape feature extraction

In this section, we conduct several experiments to test different parameters in order to obtain an effective representation for the in-air hand-drawn number and shape recognition. The accuracy rate for each experiment is shown Table A.1. Initially, we have modulated the $\mathbf{u}_0$ with a unit vector (*i.e.*, [1, 1 , ... ,1]) with length n. This is the same as taking the $\mathbf{u}_0$ on its own.

1. F_1= $\mathbf{u}_0$.
2. $F_2 = \mathbf{u}_0 * i, \qquad i = 0, 1, ..., n-1.$

   This modulation provides a higher weight towards the end of the shape.
3. $F_3 = \mathbf{u}_0 * i^2, \qquad i = 0, 1, ..., n-1.$

   This modulation provides a much higher weight towards the end of the shape.

   Looking at F_2 and F_3, one can wonder why we should give a higher weight to the end, why not at the start too. Therefore, we define two more functions F_4 and F_5.
4. $F_4 = \mathbf{u}_0 * (n - i), \qquad i = 0, 1, ..., n-1.$
5. $F_5 = F_4^2$.

   Now, we can add more two modulations using higher weight at both ends but lower weight at the centre.

6. F_6= $n - (i + 1 - (n + 1)/2) + 1, \qquad i = 0, 1, ..., n - 1.$

7. F_7= $n - (i + 1 - (n + 1)/2)^2 + 1, \qquad i = 0, 1, ..., n - 1.$

   We can also taper this feature to the centre,

8. F_8= $(i + 1 - (n + 1)/2) + 1, \qquad i = 0, 1, ..., n - 1.$

9. F_9= $(i + 1 - (n + 1)/2)^2 + 1, \qquad i = 0, 1, ..., n - 1.$

   Also, we note, some shapes are started at the top and some are started at the centre. Then, in the pre-processing step, we centre the shape with respect to the $(0, 0, 0)$ coordinate. Then, we compute the distance and the angle with respect to the $(0, 0, 0)$. Two features are tested as followed,

10. $F_{10} = \mathbf{u}_0 * \hat{\mathbf{r}}$.

11. $F_{11} = \mathbf{u}_0 * \theta$.

    Finally, we use features combination based on$\hat{\mathbf{r}}$, θ, and the graph eigenvectors of the combinatorial and the normalised versions as follows,

12. F_{12} = $\mathbf{U_L} * \hat{\mathbf{r}}$.

13. F_{13} = $\mathbf{U_L} * \theta$.

14. F_{14} = $\mathbf{U}_{\mathcal{L}} * \hat{\mathbf{r}}$.

15. F_{15} = $\mathbf{U}_{\mathcal{L}} * \theta$.

    Based on these results, we test a different set of features as follows,

16. F_{16} = Concatenating (F_2+F_{10}+F_{11}).

17. F_{17} = Concatenating (F_1+F_2+F_3+F_{10}+F_{11}).

18. F_{18} = Concatenating (F_3+F_{10}+F_{11}).

19. F_{19} = Concatenating (F_1+F_3+F_{10}+F_{11}).

20. F_{20} = Concatenating (F_{10}+F_{11}).

21. F_{21} = Concatenating (F_9+F_{10}+F_{11}).

Table A.1: Accuracy rate (%) of various features of numbers and shapes.

Exp. no.	**Feature combination**	**Accuracy of numbers**	**Accuracy of shapes**
1	F_1	96.8	94.6
2	F_2	93.7	93.4
3	F_3	98.01	96.2
4	F_4	95.2	93.8
5	F_5	95.1	93.3
6	F_6	93.8	91.9
7	F_7	96.3	94.4
8	F_8	94.01	91.5
9	F_9	98.01	96.2
10	F_{10}	95.7	94.2
11	F_{11}	96.1	94
12	F_{12}	55.56	51.9
13	F_{13}	56.03	52.31
14	F_{14}	74.34	71.65
15	F_{15}	64.65	61.8
16	F_{16}	99.27	98.25
17	F_{17}	99.18	98
18	F_{18}	99.42	99.37
19	F_{19}	99.1	98.8
20	F_{20}	99.04	98
21	F_{21}	99.56	99.44

Appendix B

In-air hand-drawn number and shape feature visualisation

In this section, we illustrate in details the proposed features for in-air hand-drawing numbers and shapes recognition. For all figures in this section, please follow the legend in Figure. B.1. X-axes refer to the index of the feature and Y-axes represent the value of the features. For comparison purposes, we show all these features together (Figure. B.22 and Figure. B.23) at the end of this section.

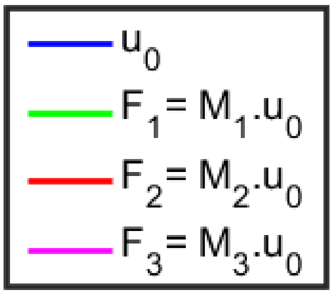


Figure B.1: Legend of the figures in this section.

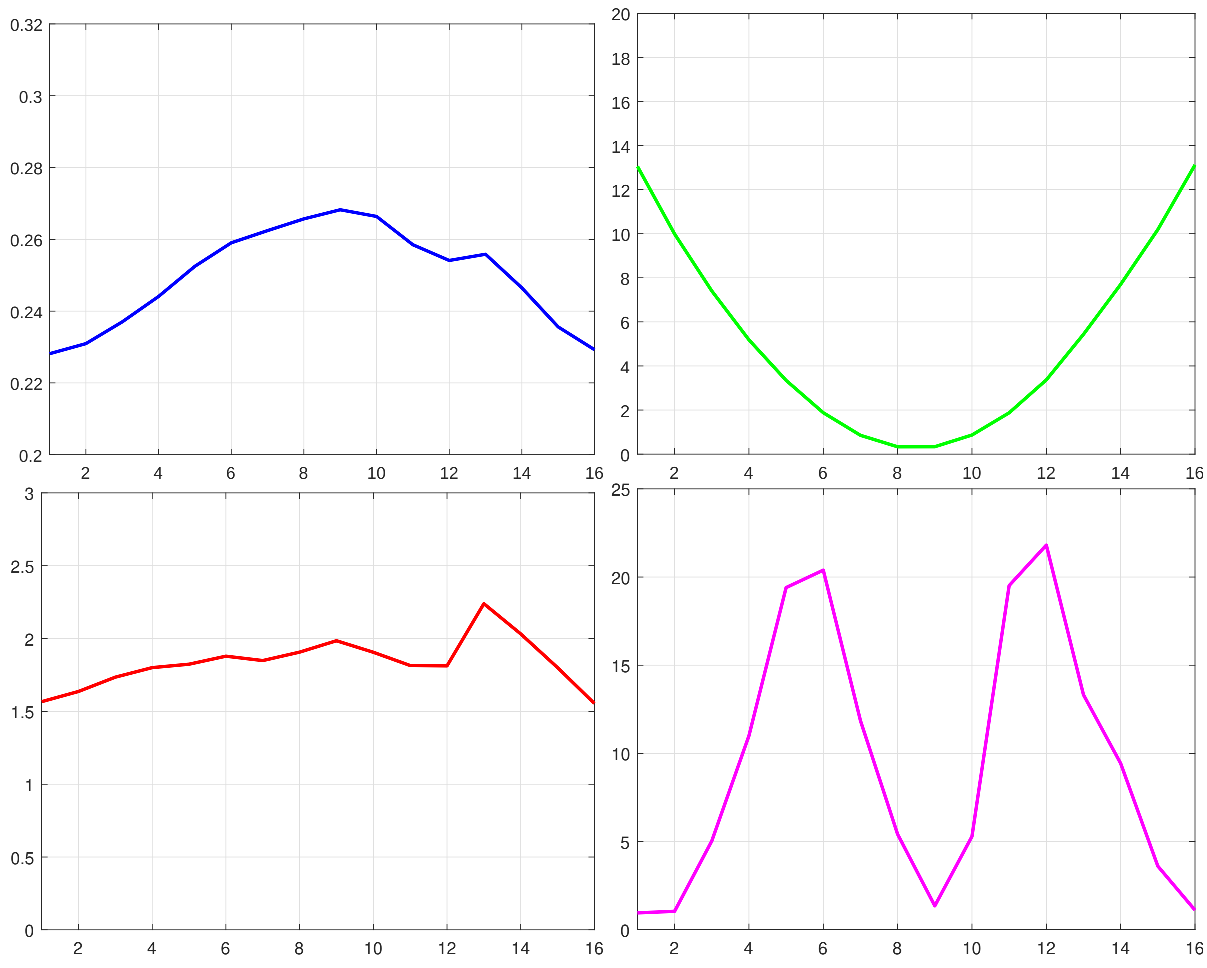


Figure B.2: Features of number 0.

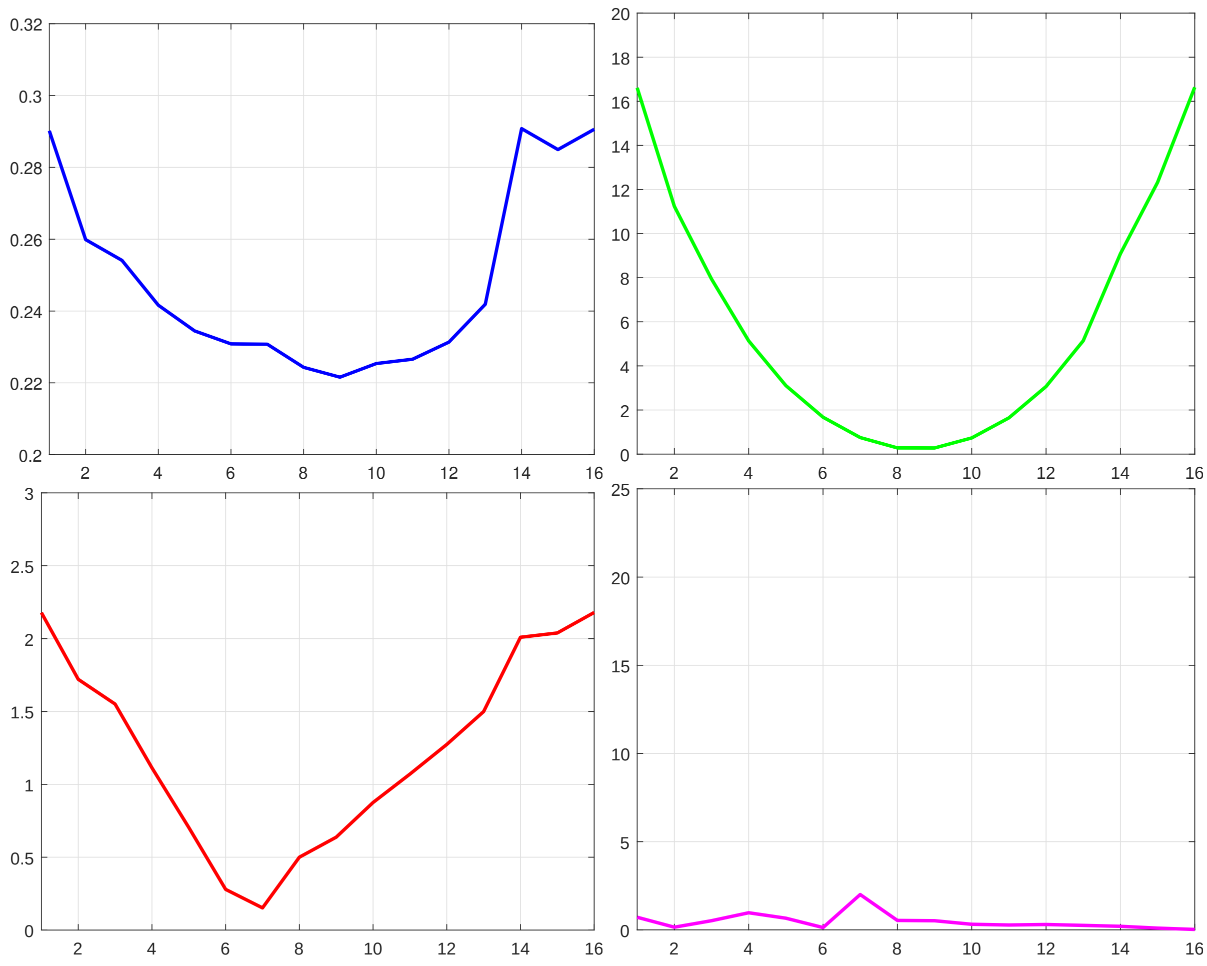


Figure B.3: Features of number 1.

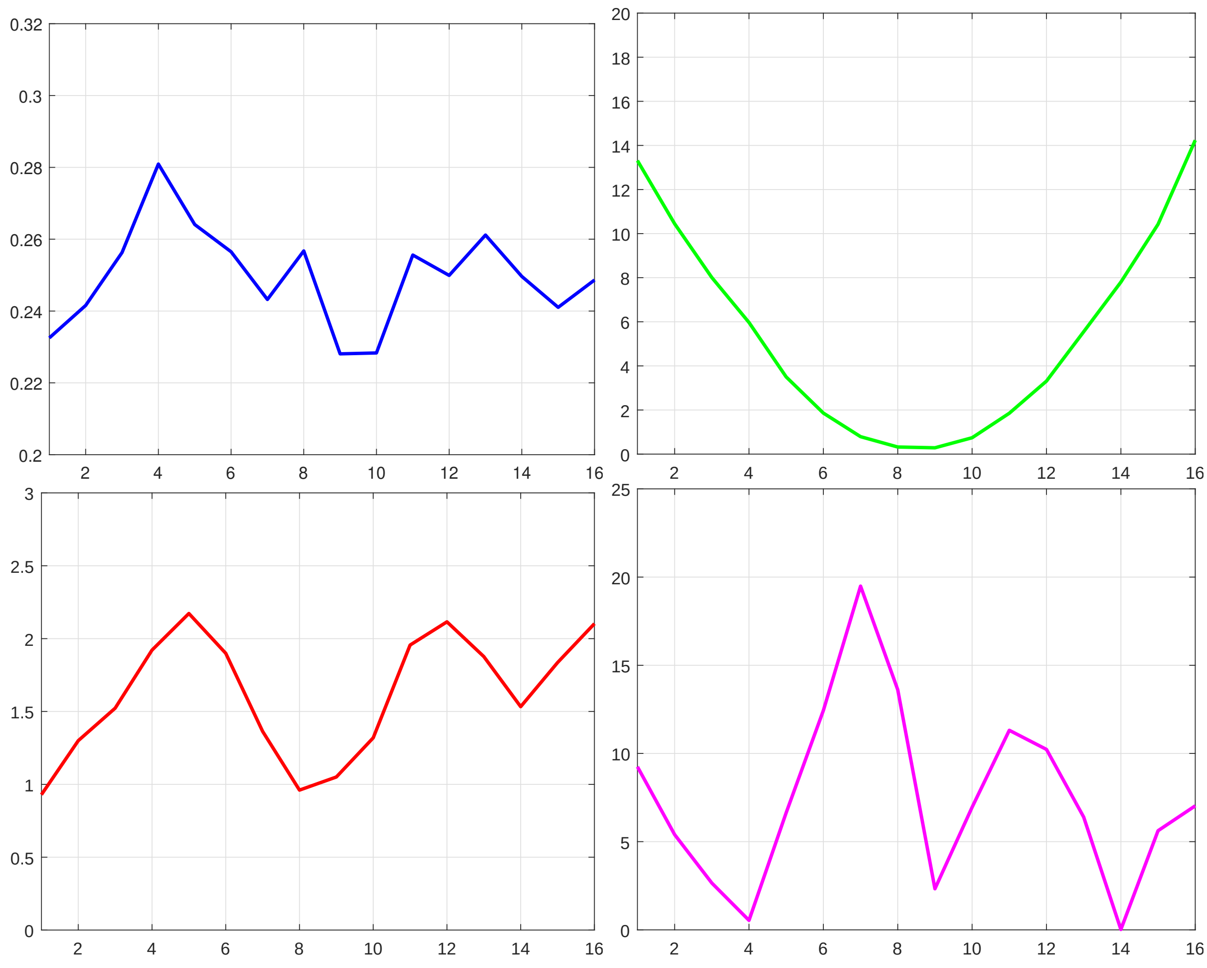


Figure B.4: Features of number 2.

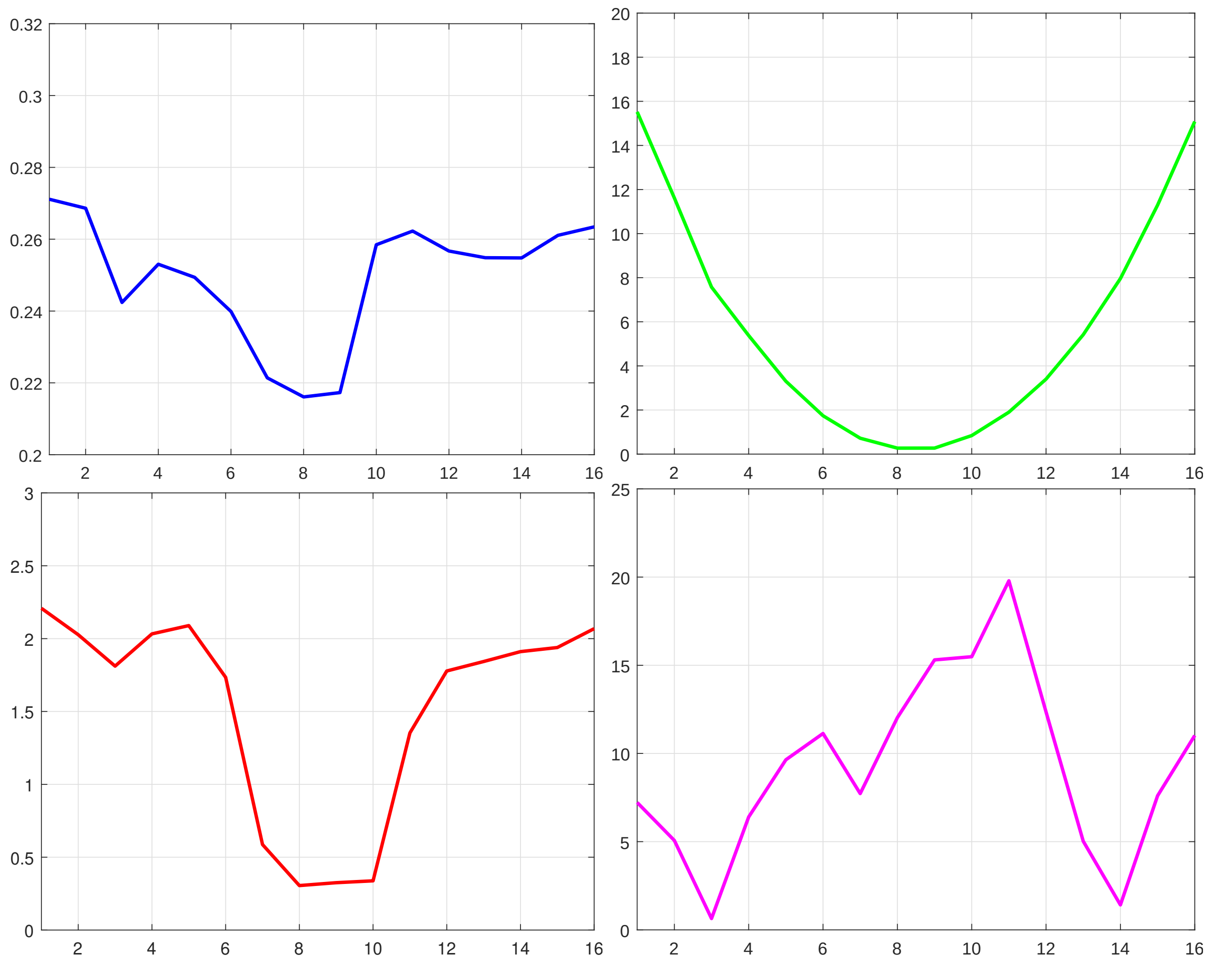


Figure B.5: Features of number 3.

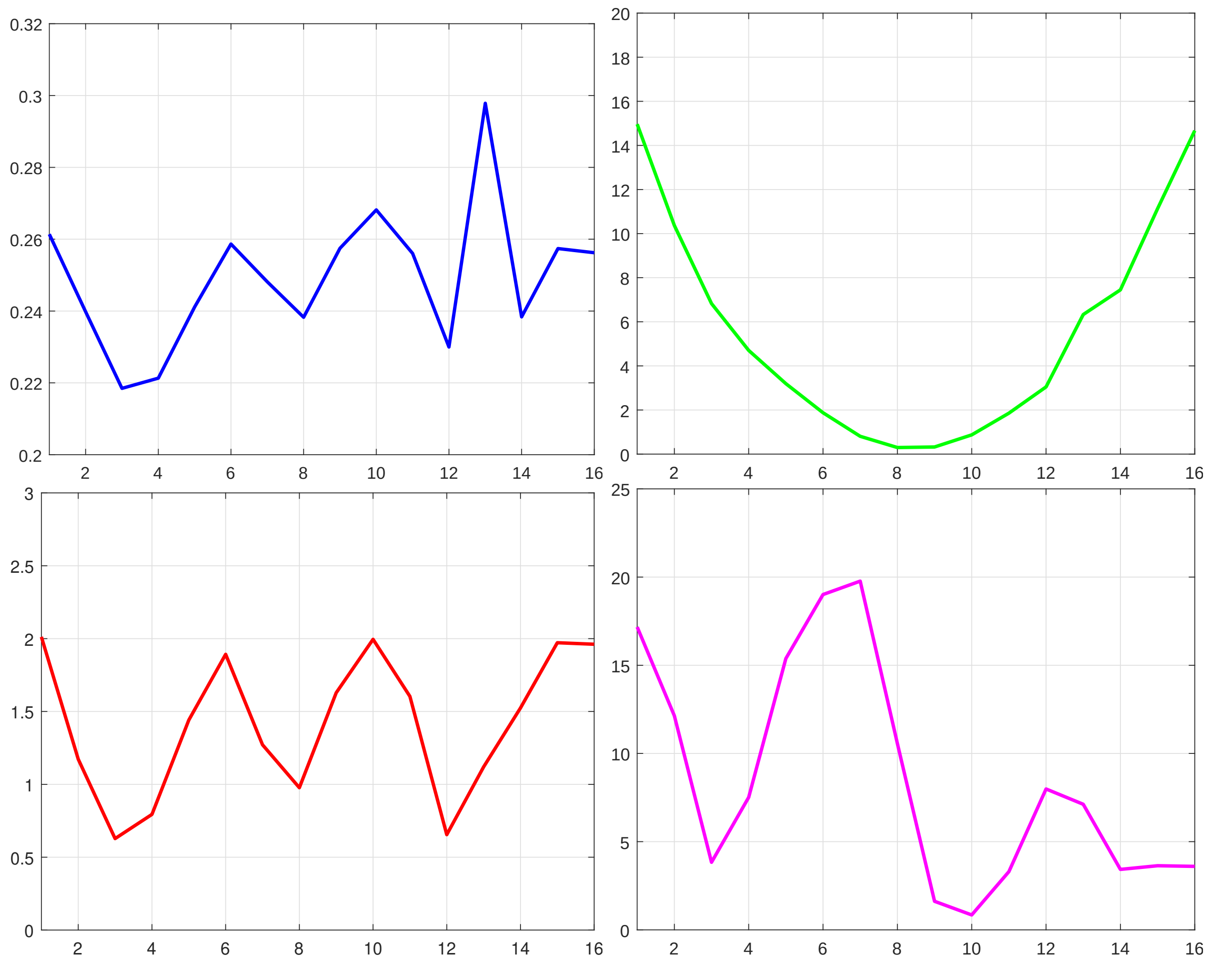


Figure B.6: Features of number 4.

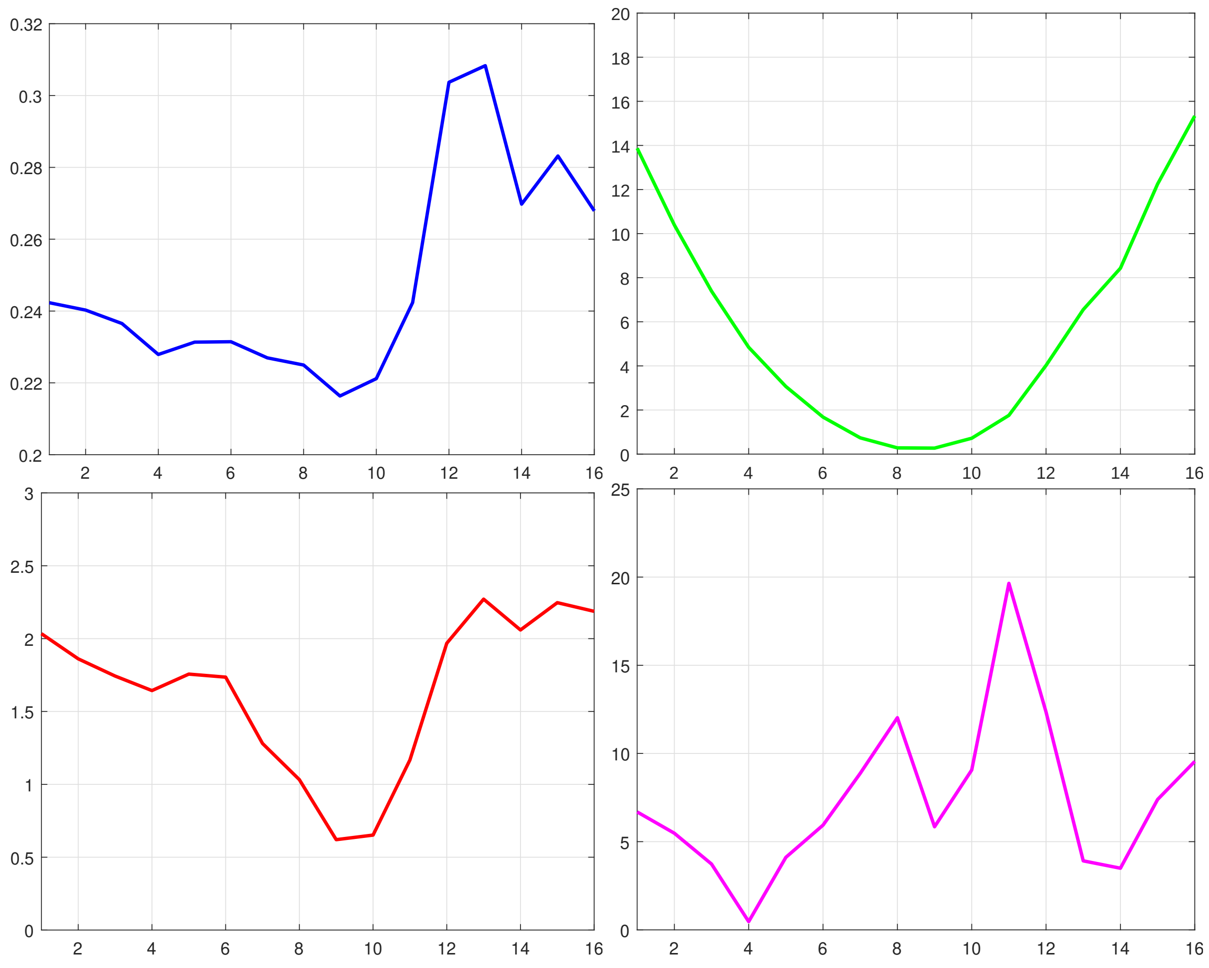


Figure B.7: Features of number 5.

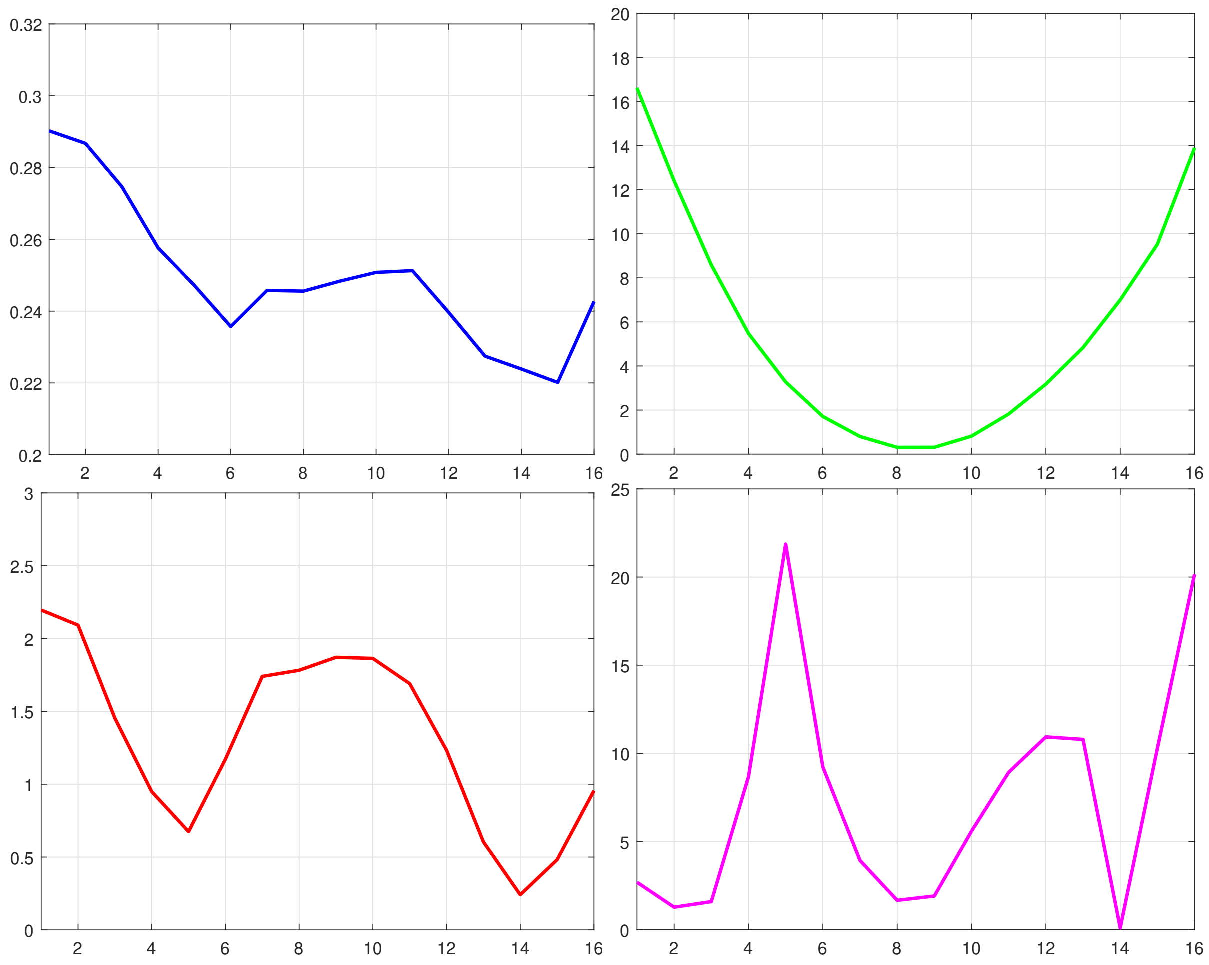


Figure B.8: Features of number 6.

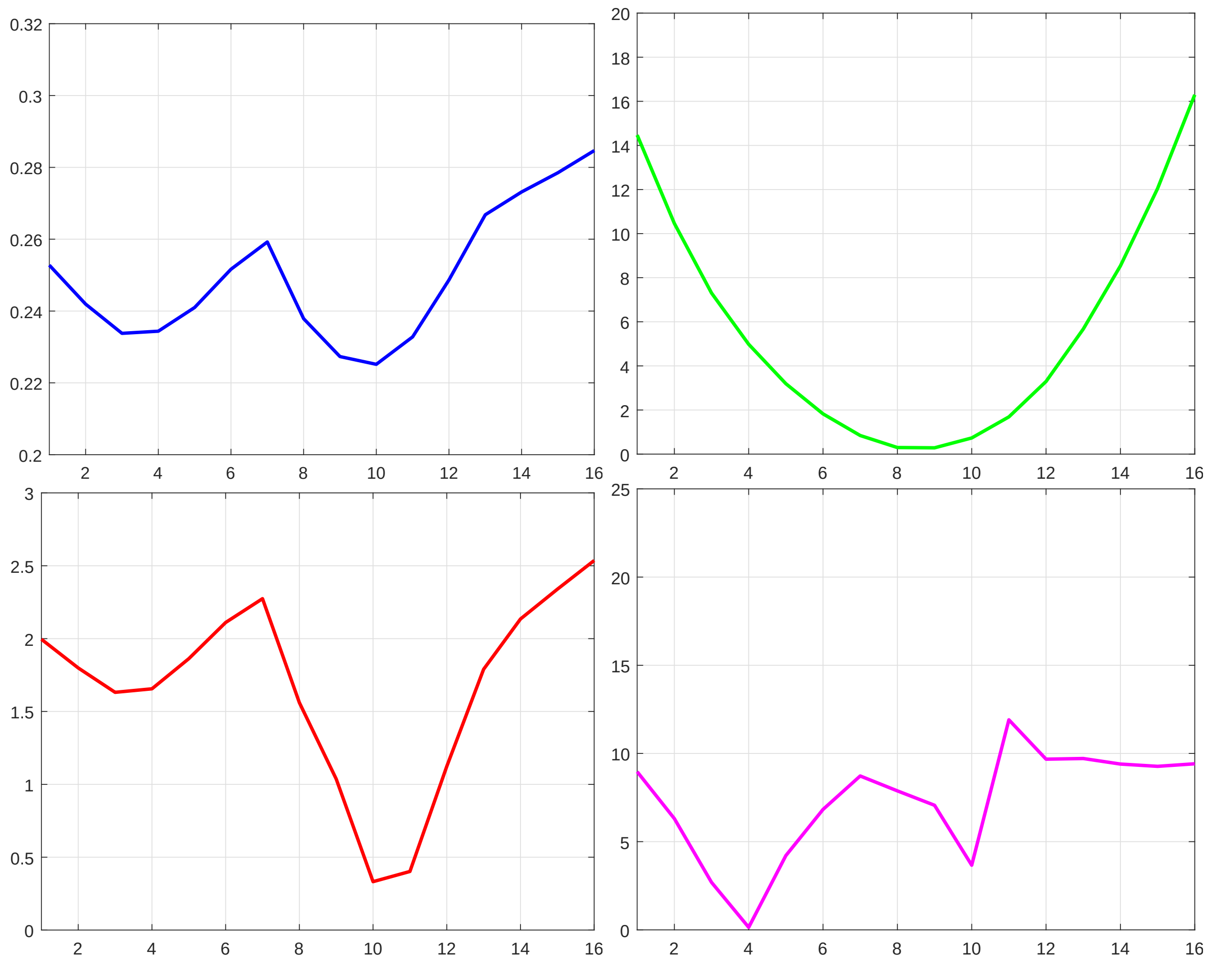


Figure B.9: Features of number 7.

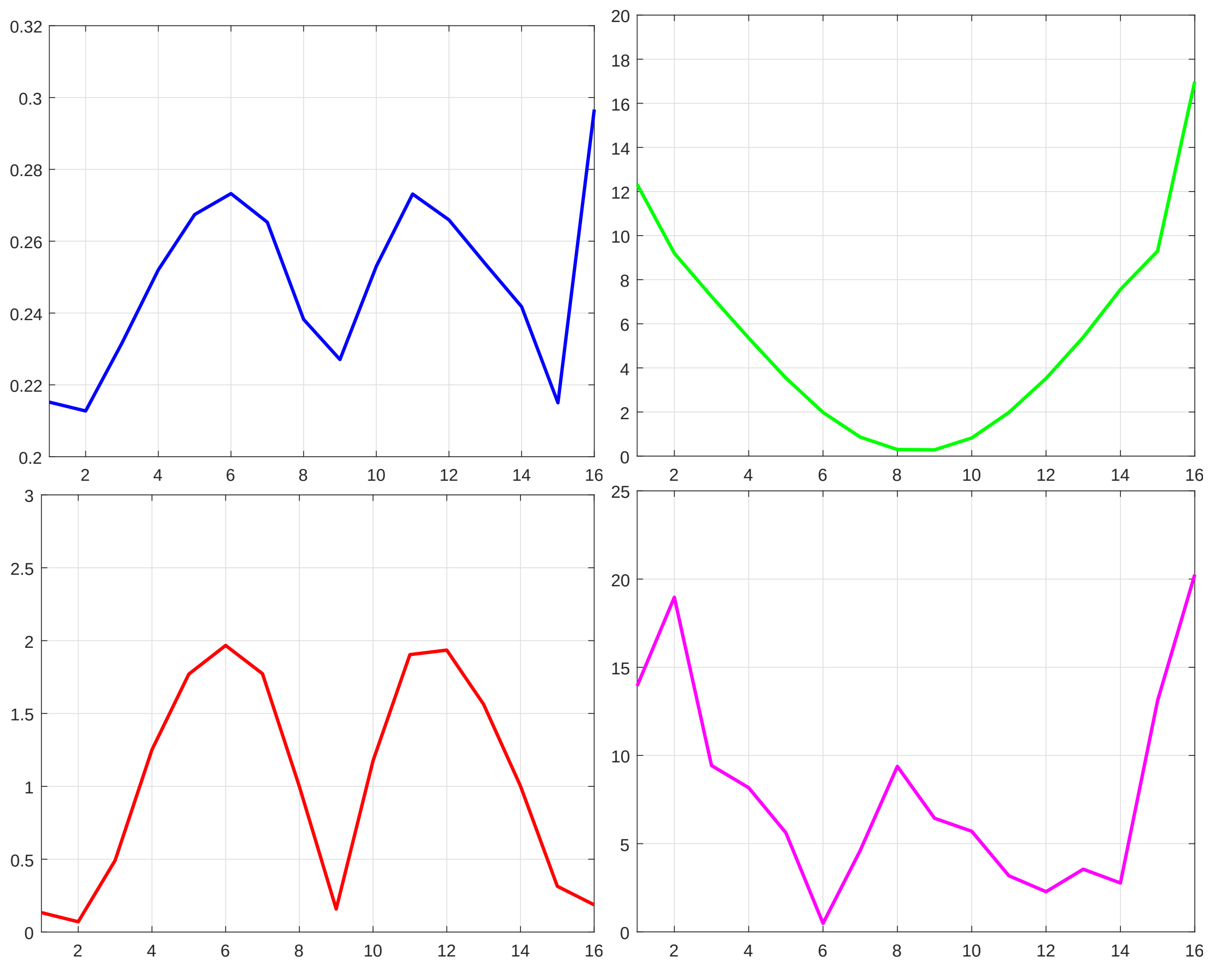


Figure B.10: Features of number 8.

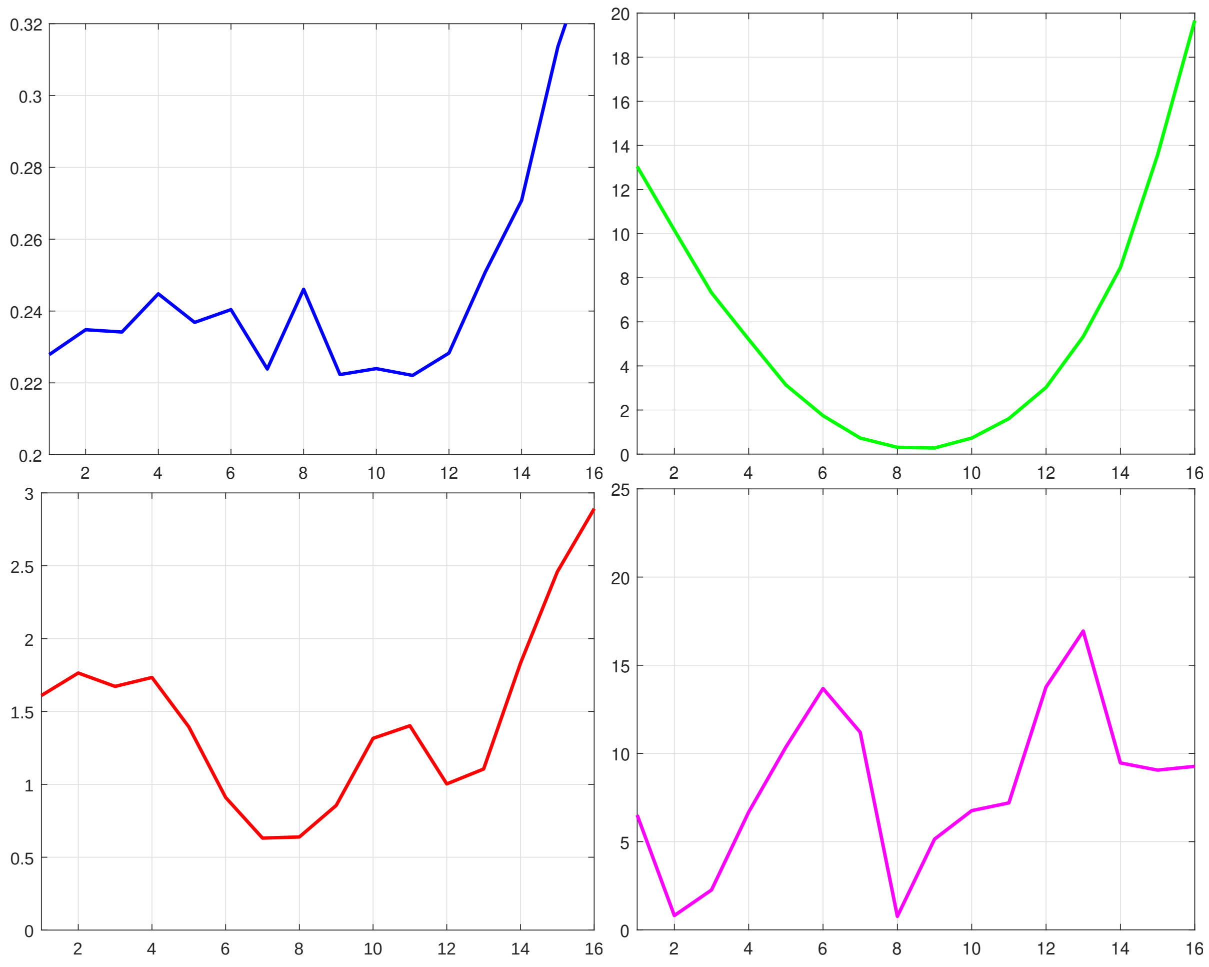


Figure B.11: Features of number 9.

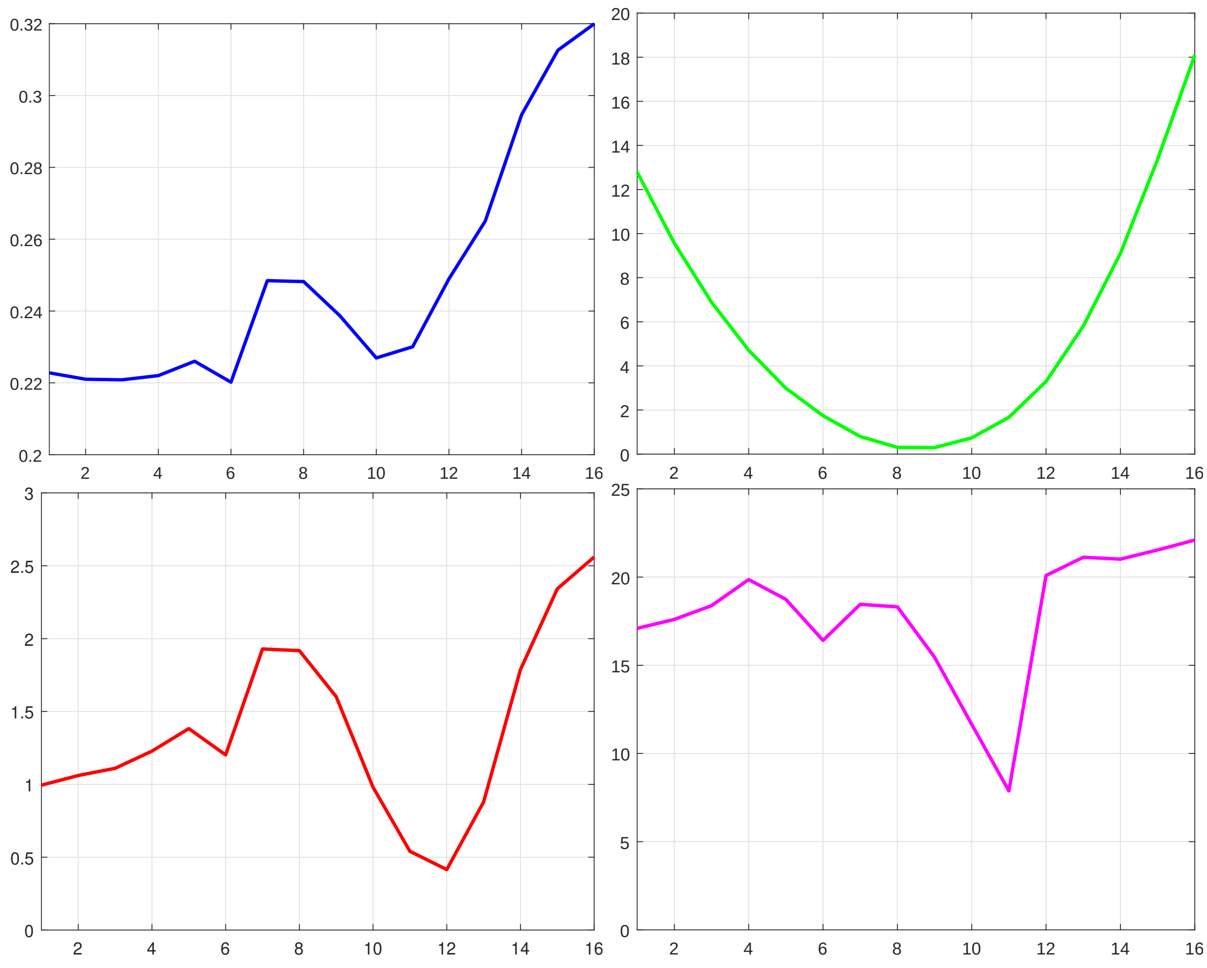


Figure B.12: Features of shape **Yes**.

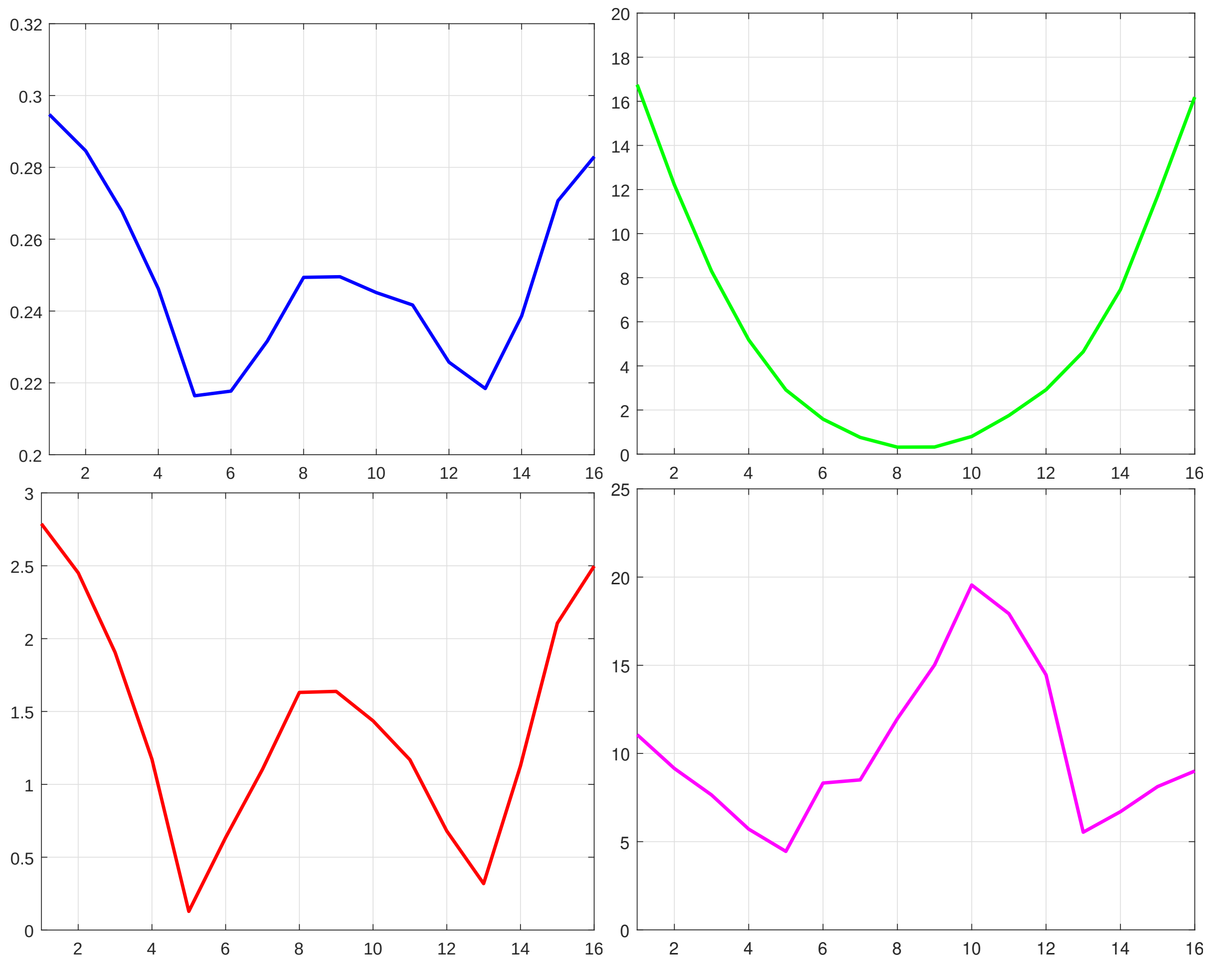


Figure B.13: Features of shape **No**.

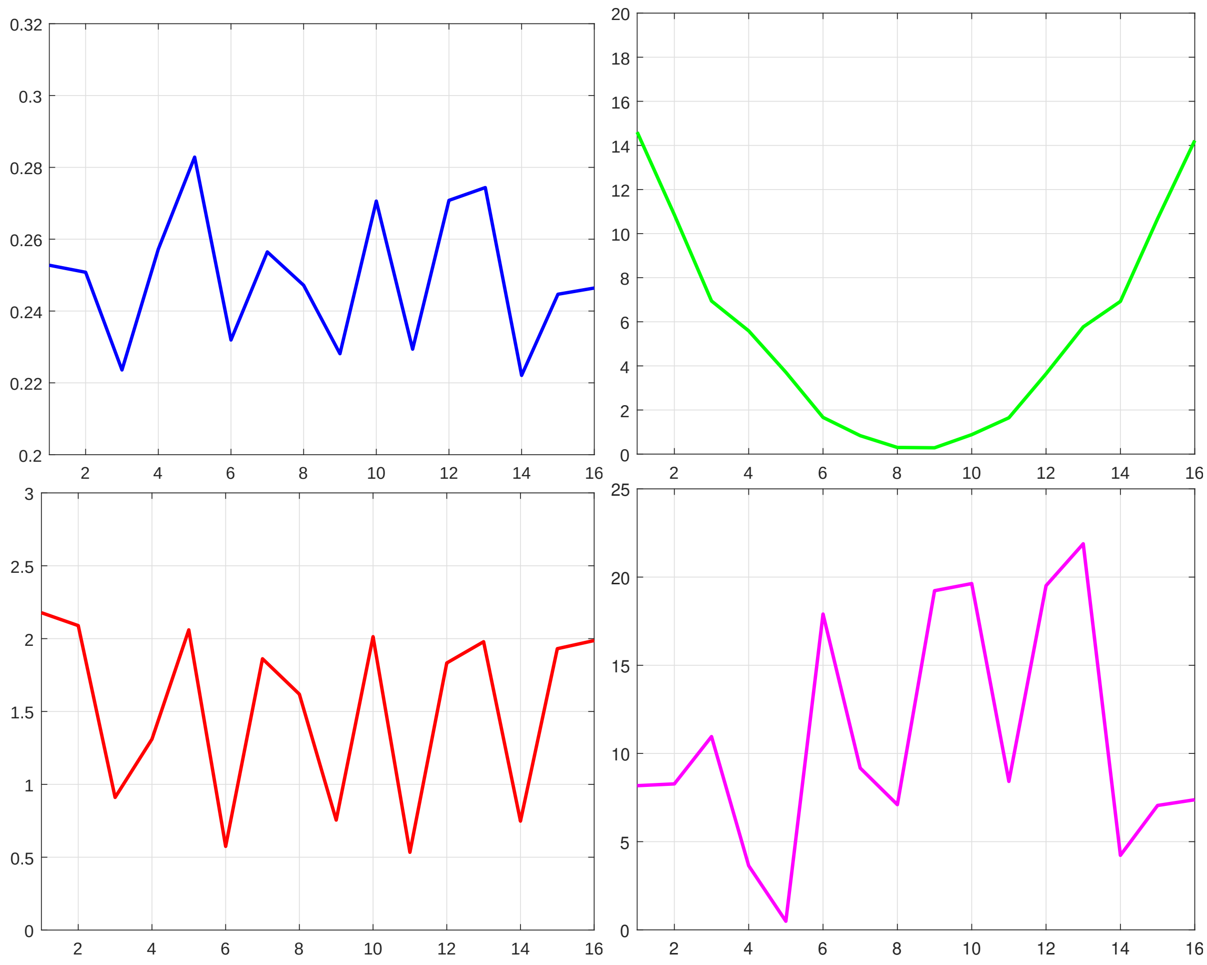


Figure B.14: Features of shape **Star**.

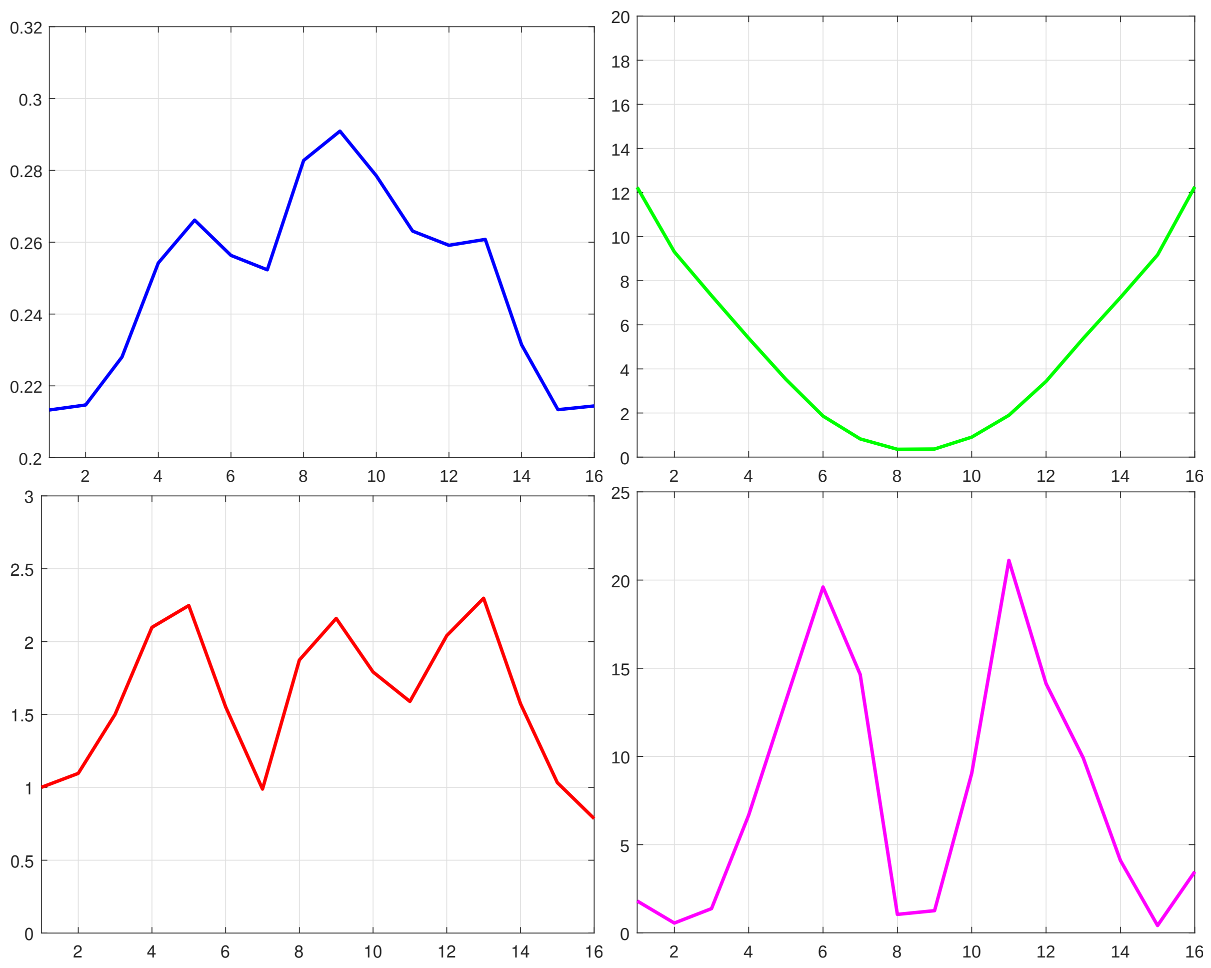


Figure B.15: Features of shape **Heart**.

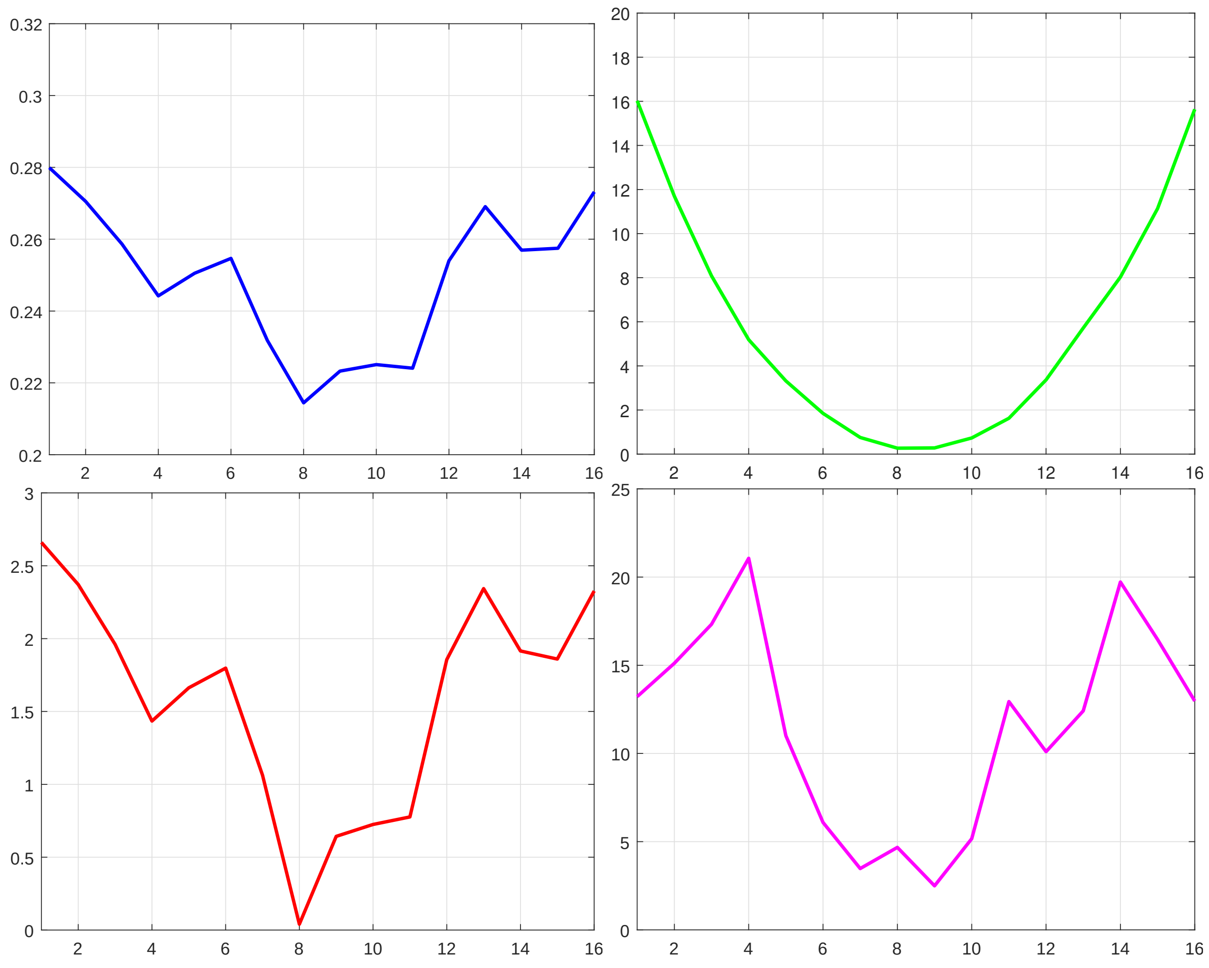


Figure B.16: Features of shape **M**.

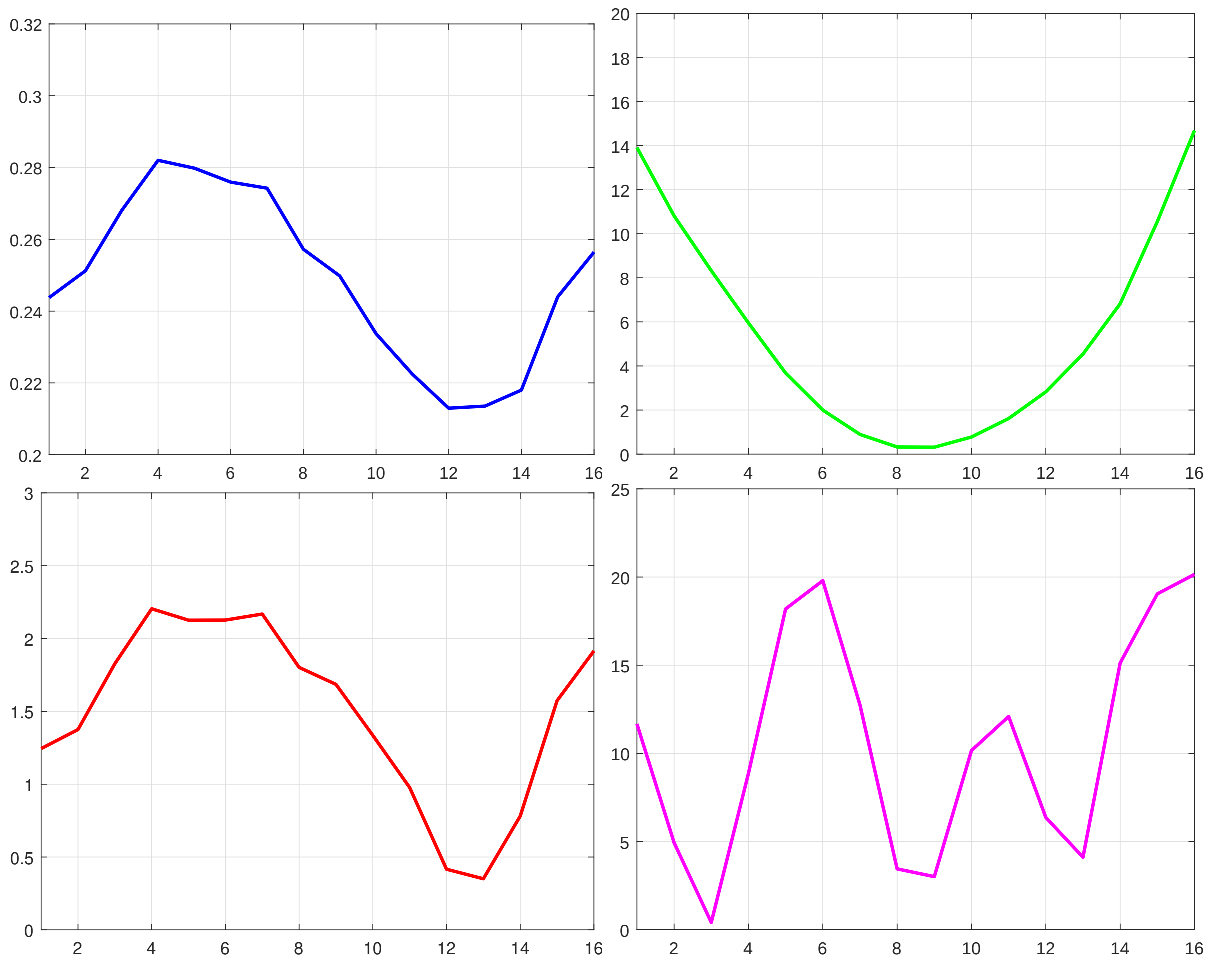


Figure B.17: Features of shape **G**.

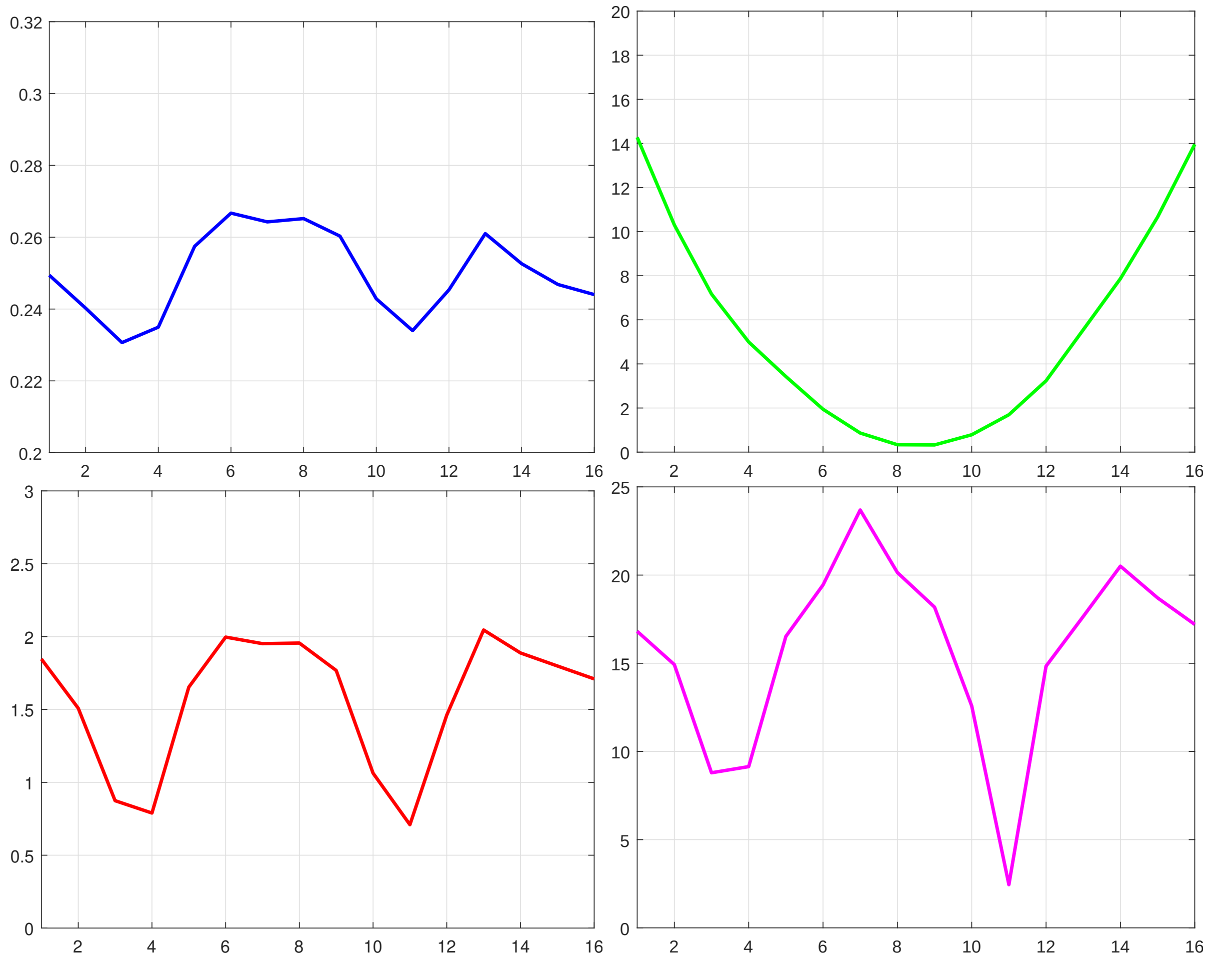


Figure B.18: Features of shape **Rectangular**.

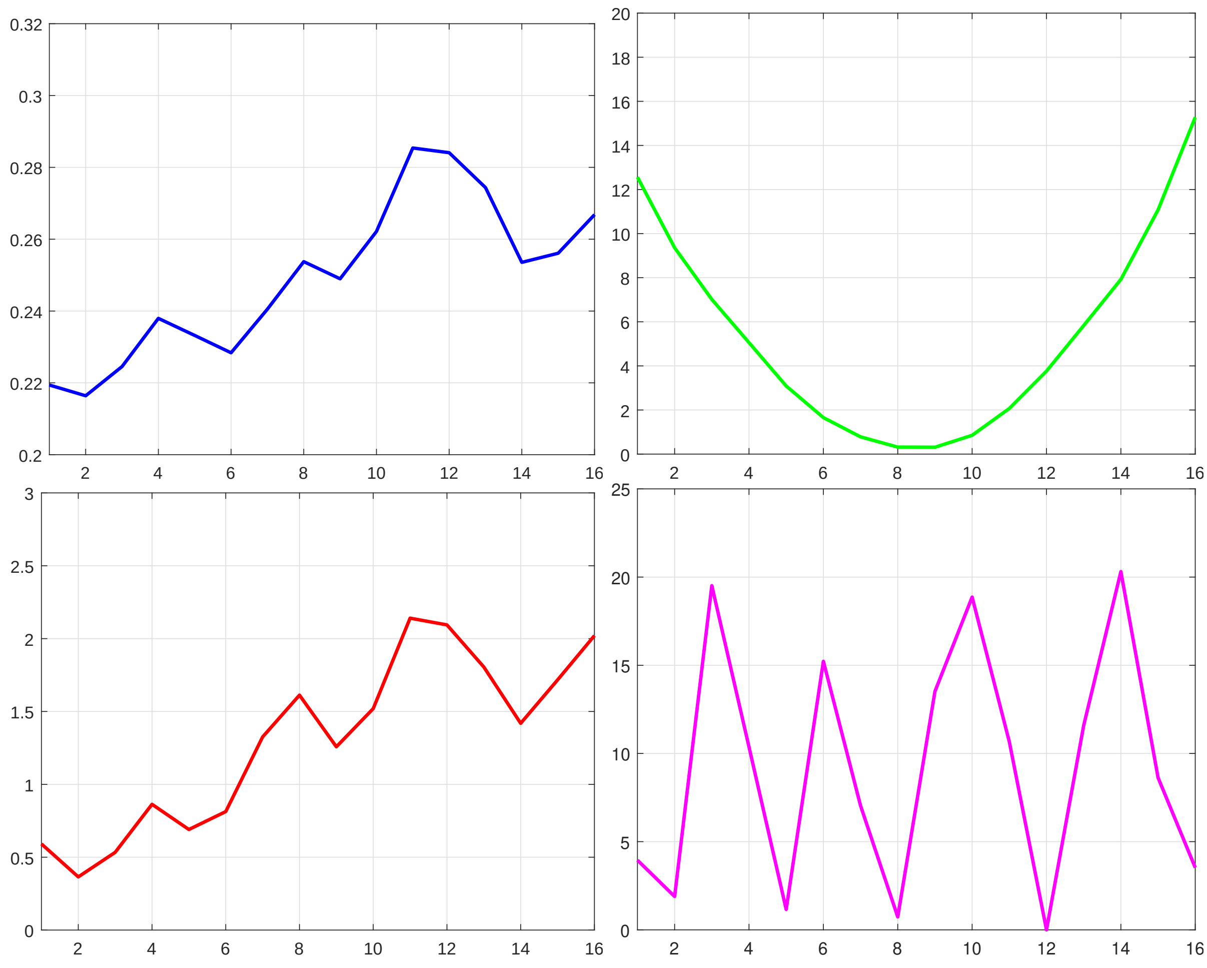


Figure B.19: Features of shape **Spiral**.

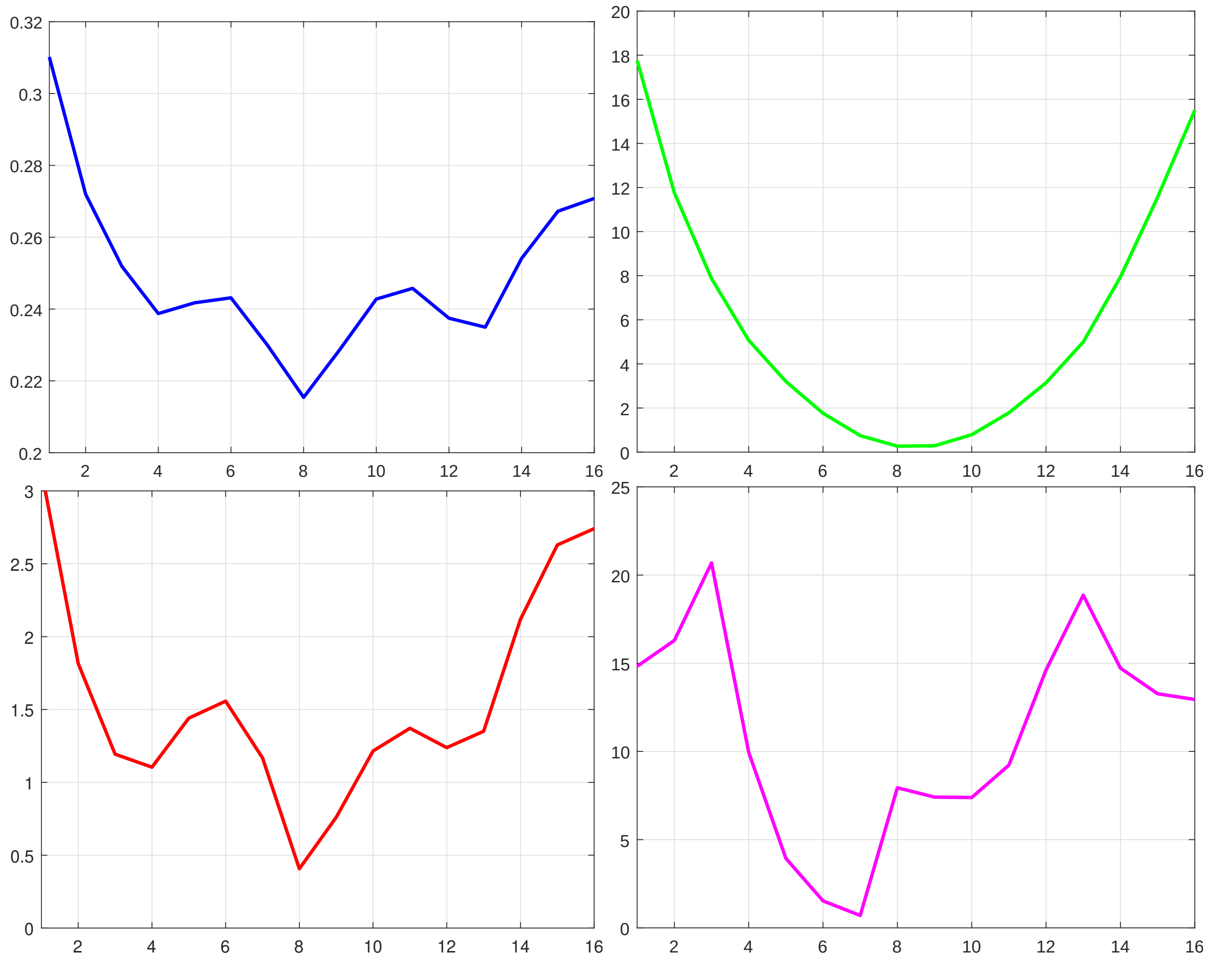


Figure B.20: Features of shape **N**.

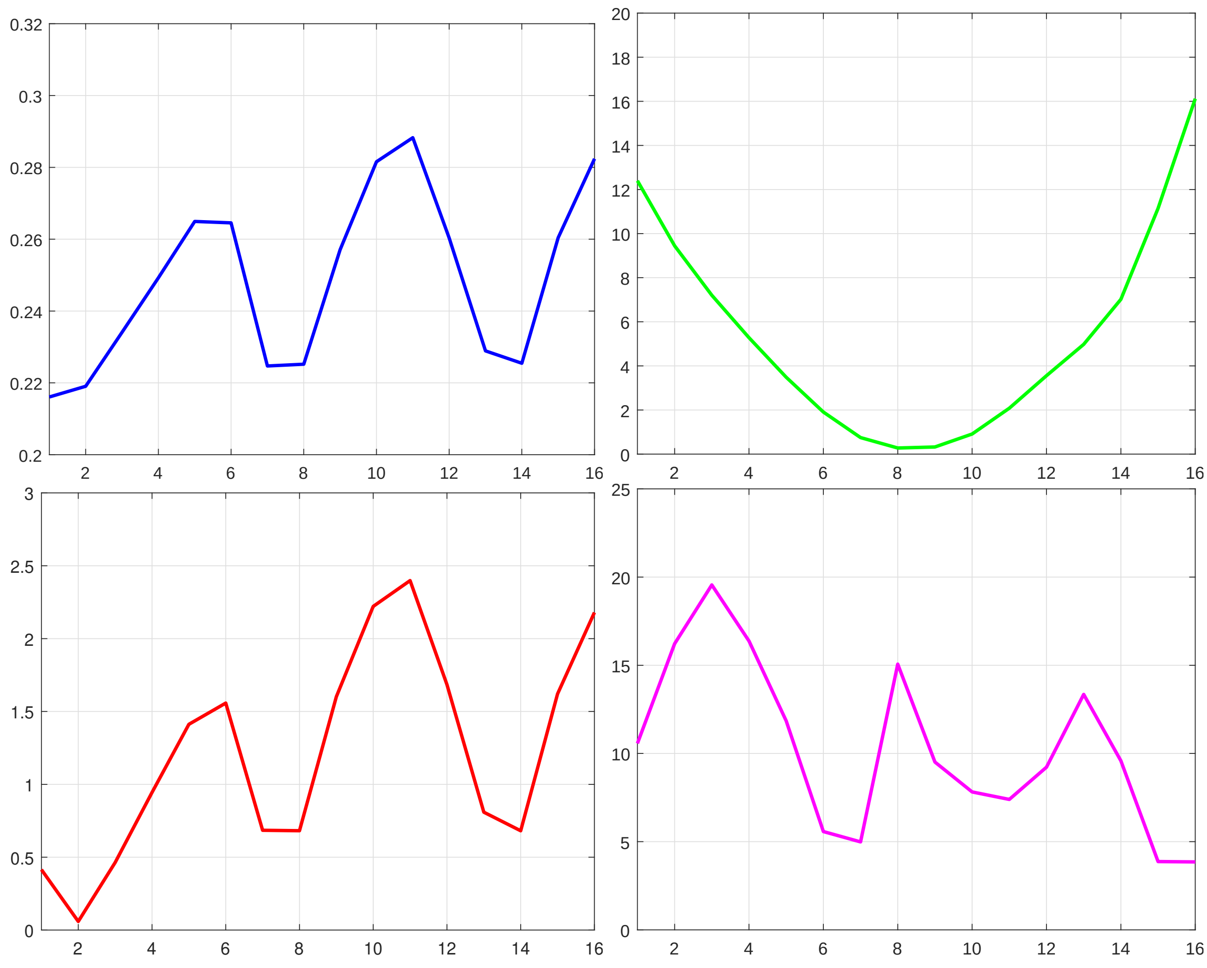


Figure B.21: Features of shape **d**.

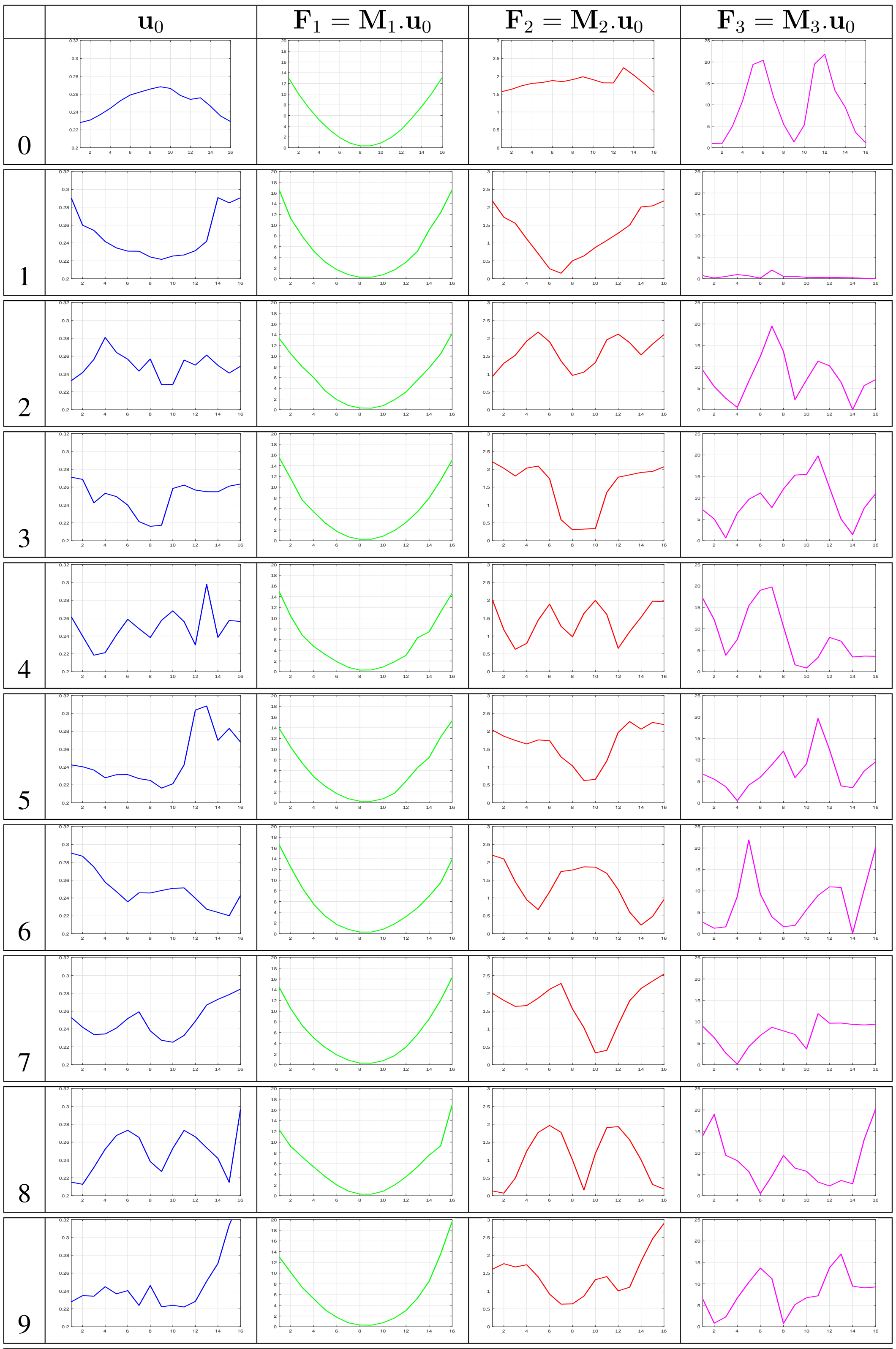


Figure B.22: The proposed feature vectors for the numbers 0 to 9 (shown in each row): Column 1: Shows the eigenvector u_0; Column 2 - Column 4 show the feature vector parts $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$, respectively.

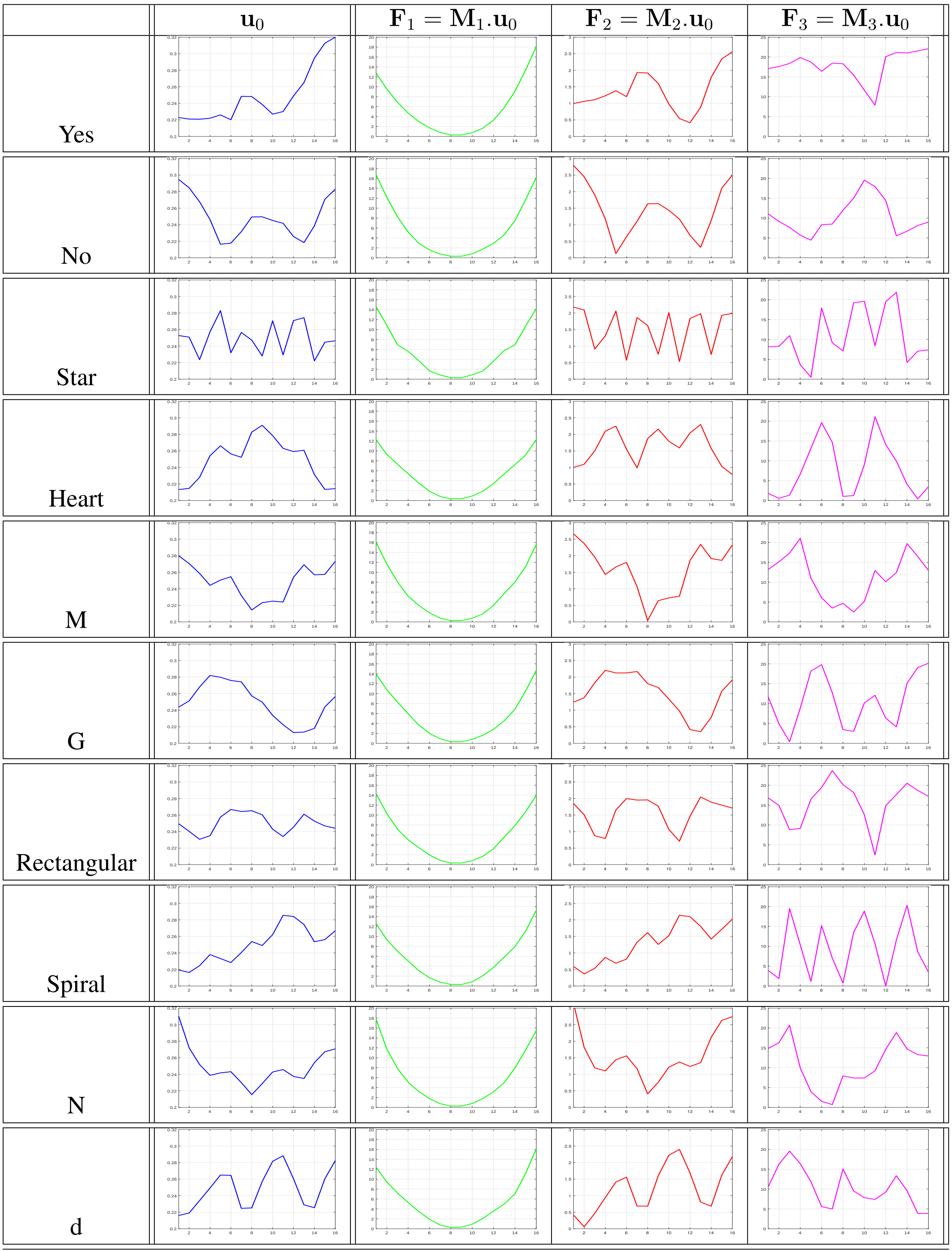


Figure B.23: The proposed feature vectors for the shapes (shown in each row): Column 1: Shows the eigenvector $\mathbf{u}_0$; Column 2 - Column 4 show the feature vector parts $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$, respectively.

Appendix C

Examples of spectral partitioning

In this section, we show more samples for the partitioning methods

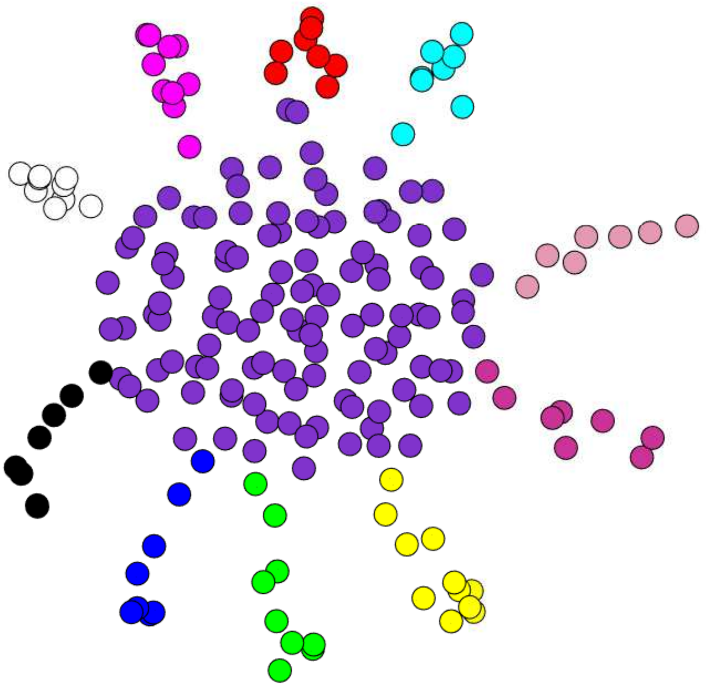


Figure C.1: Samples of the proposed partitioning method-Crabs.

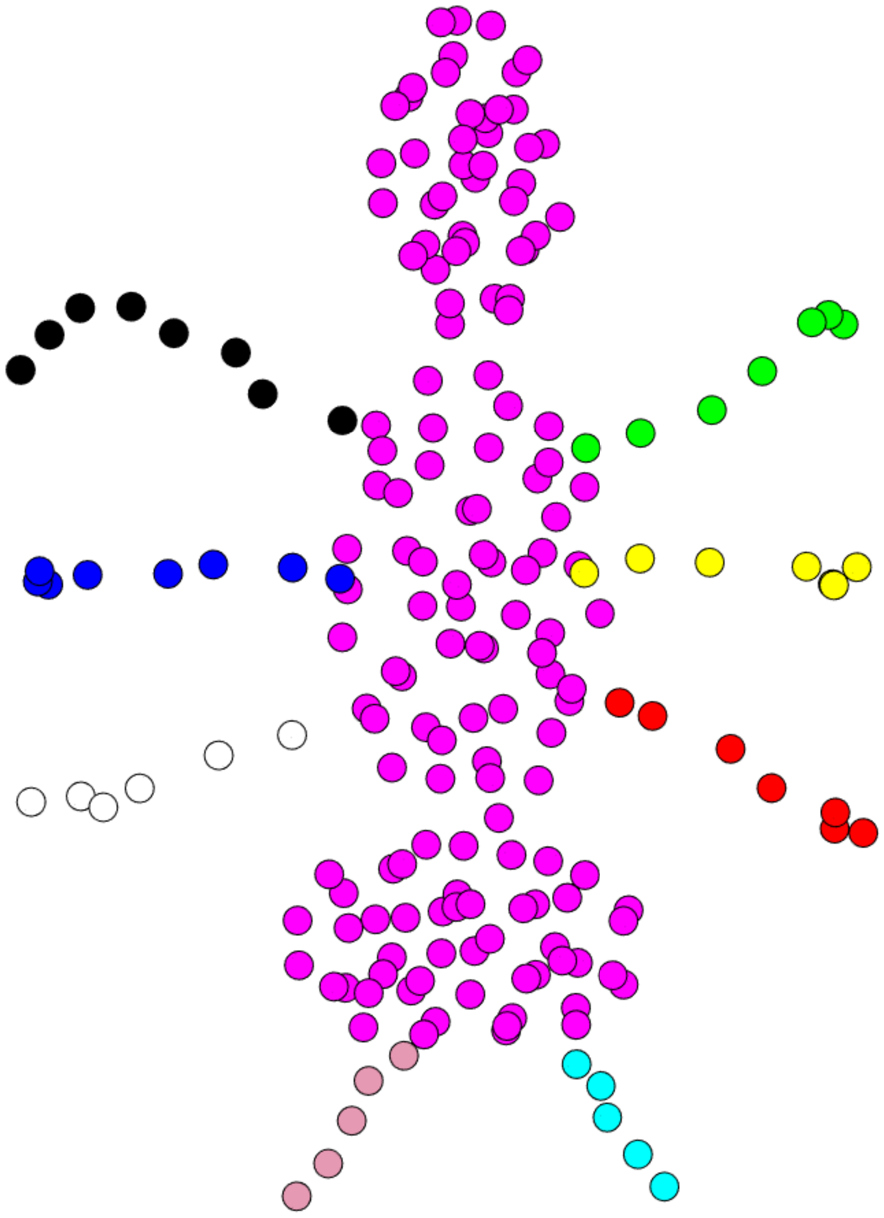


Figure C.2: Samples of the proposed partitioning method-Ants.

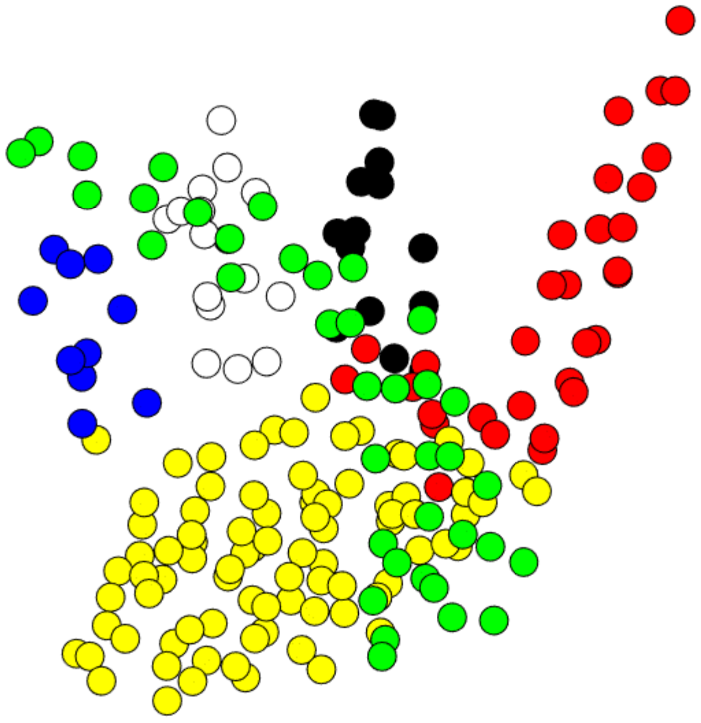


Figure C.3: Samples of the proposed partitioning method-Hand.

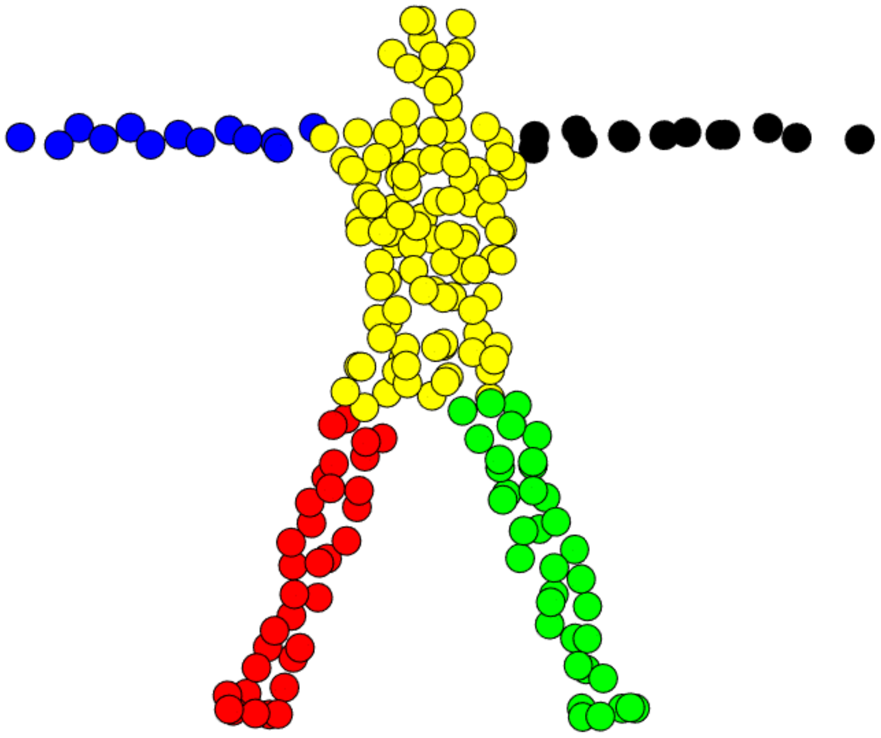


Figure C.4: Samples of the proposed partitioning method-Human.

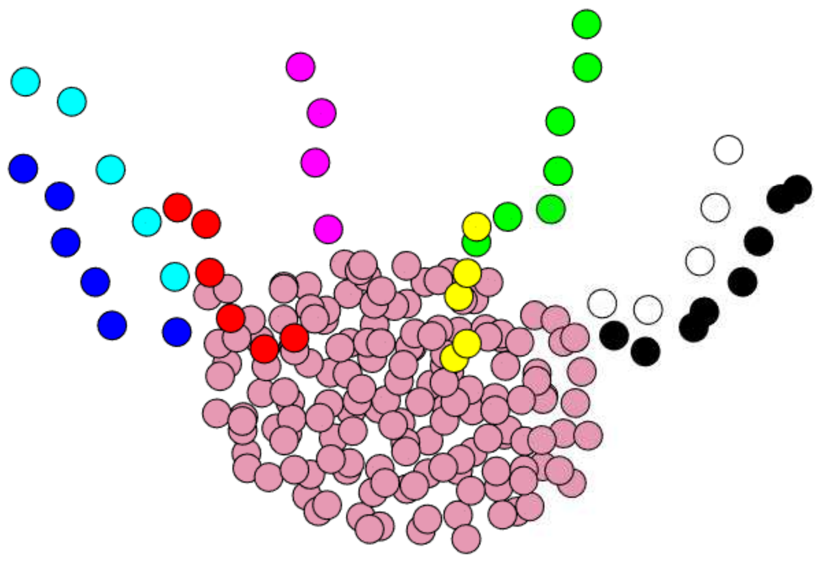


Figure C.5: Samples of the proposed partitioning method-Octopus.

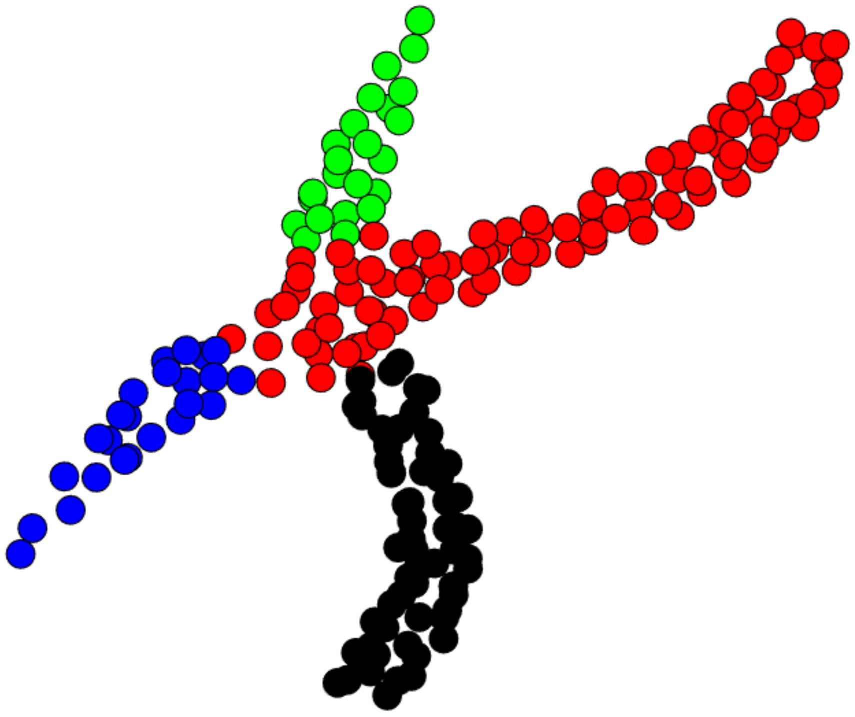


Figure C.6: Samples of the proposed partitioning method-Pliers.

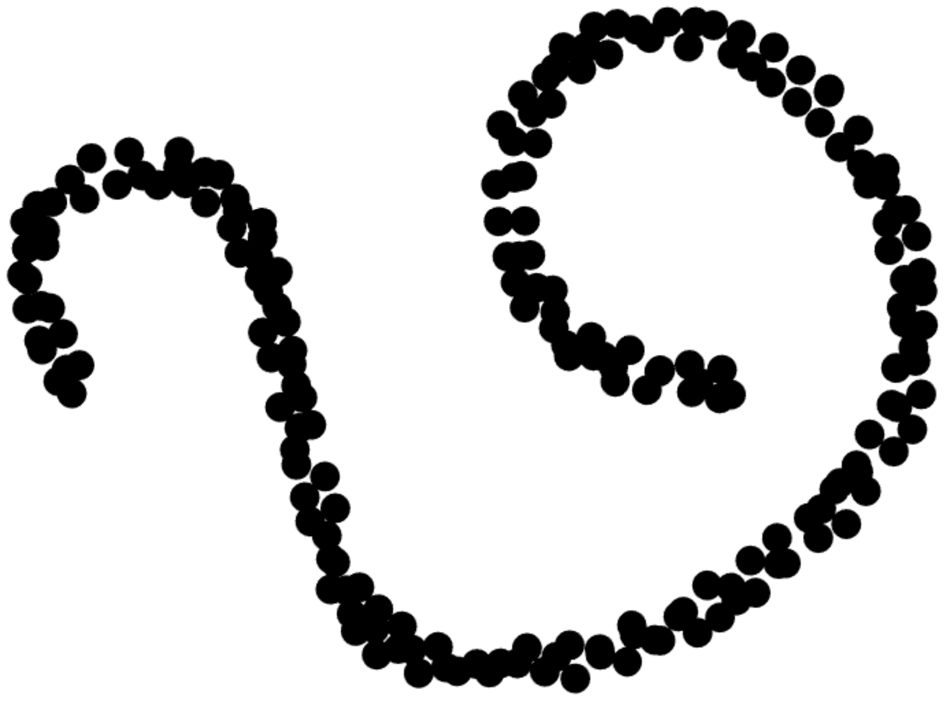


Figure C.7: Samples of the proposed partitioning method-Snake.

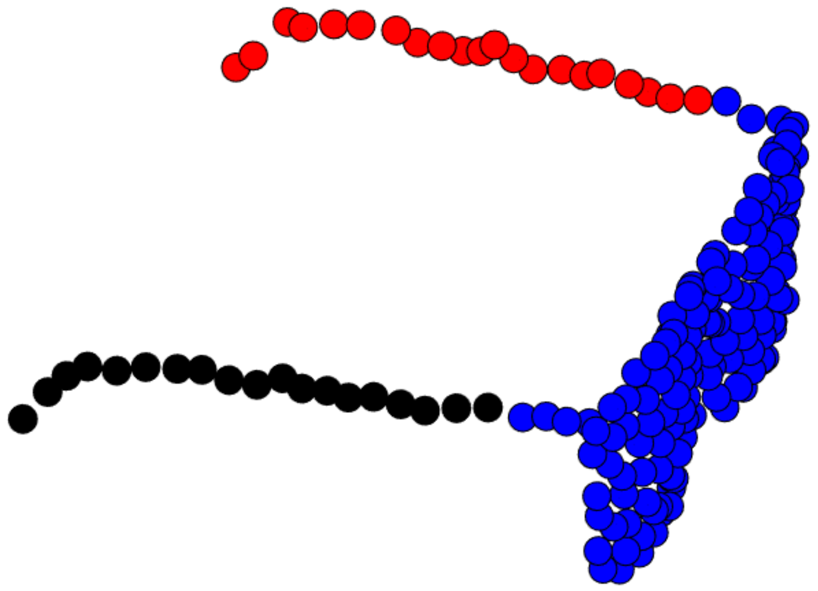


Figure C.8: Samples of the proposed partitioning method-Spectacle.

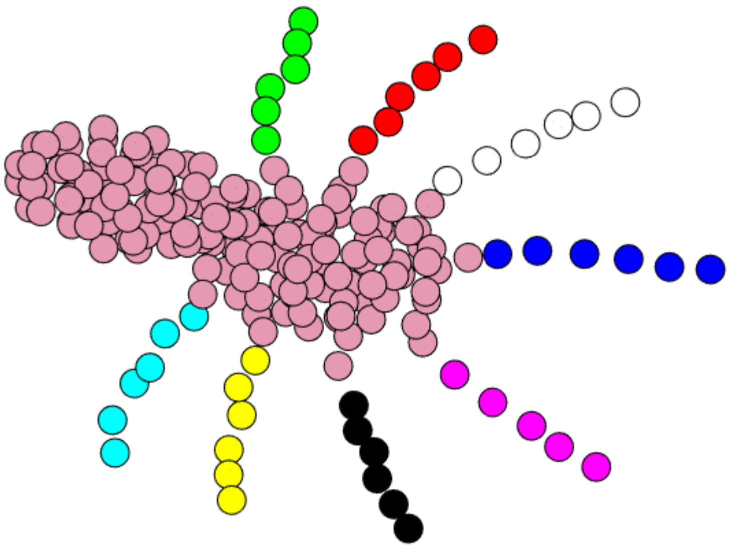


Figure C.9: Samples of the proposed partitioning method-Spider.

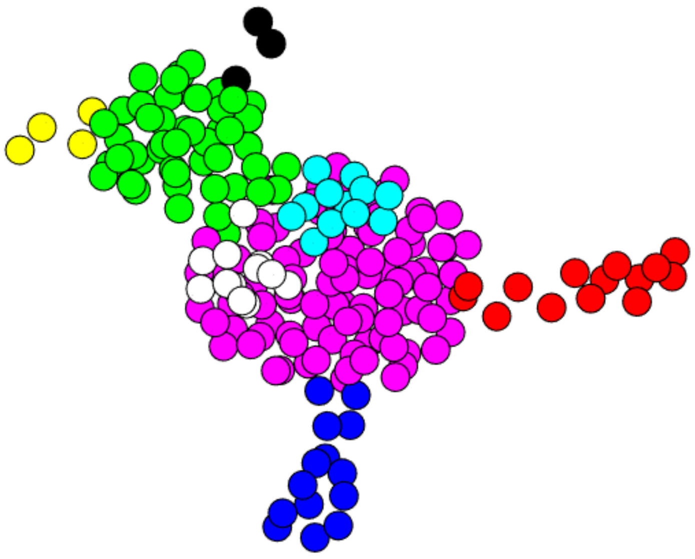


Figure C.10: Samples of the proposed partitioning method-Teddy.

Bibliography

[1] P. L. L., "Lecture on graph mining," September 2017. [Online]. Available: https://www.slideshare.net/pierluca.lanzi/dmtm-lecture-18-graph-mining

[2] R. B. Dan and P. Mohod, "Survey on hand gesture recognition approaches," *structure*, vol. 15, p. 17, 2014.

[3] C. Li and A. B. Hamza, "A multiresolution descriptor for deformable 3D shape retrieval," *The Visual Computer*, vol. 29, no. 6-8, pp. 513–524, 2013.

[4] L. E. Carvalho and A. von Wangenheim, "3D object recognition and classification: a systematic literature review," *Pattern Analysis and Applications*, pp. 1–50, 2019.

[5] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[6] P. K. Saha, G. Borgefors, and G. S. di Baja, "A survey on skeletonization algorithms and their applications," *Pattern Recognition Letters*, vol. 76, pp. 3–12, 2016.

[7] Y. Cai, S. Osman, M. Sharma, M. Landis, and S. Li, "Multi-modality vertebra recognition in arbitrary views using 3D deformable hierarchical

model,” *in IEEE Transactions on Medical Imaging*, vol. 34, pp. 1676–1693, 2015.

[8] J. Lichtenauer, E. Hendriks, and M. Reinders, “Sign language recognition by combining statistical DTW and independent classification,” *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2040–2046, 2008.

[9] K. J. Peter, G. G. S. Glory, S. Arguman, G. Nagarajan, V. Devi, and K. S. Kannan, “Improving ATM security via face recognition,” in *Proc. of International Conference on Electronics Computer Technology*, vol. 6, April 2011, pp. 373–376.

[10] 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.

[11] 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.

[12] 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.

[13] R. Watt and D. Andrews, “Contour curvature analysis: hyperacuities in the discrimination of detailed shape,” *Vision research*, vol. 22, no. 4, pp. 449–460, 1982.

[14] G. Xu and W. Fang, “Shape retrieval using deep autoencoder learning representation,” in *Proc. of Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP).* IEEE, 2016, pp. 227–230.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. of Advances in neural information processing systems*, 2016, pp. 3844–3852.

[20] S. Jianbo and J. Malik, "Normalized cuts and image segmentation," *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[21] K. Riesen, "Structural pattern recognition with graph edit distance," *Advances in computer vision and pattern recognition, Cham*, 2015.

[22] M. Pavan and M. Pelillo, "A new graph-theoretic approach to clustering and segmentation," in *Proc. of Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2003, pp. I–I.

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

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

[25] 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.

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

[27] F. Zhou and F. De La Torre, “Factorized graph matching,” *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 99, pp. 1–1, 2015.

[28] S. Shapiro and J. Brady, “Feature-based correspondence: an eigenvector approach,” *Image and vision computing*, vol. 10, no. 5, pp. 283–288, 1992.

[29] B. W. Miners, O. A. Basir, and M. S. Kamel, “Understanding hand gestures using approximate graph matching,” *in IEEE Transactions on Systems, Man, and Cybernetics: Systems and Humans*, vol. 35, no. 2, pp. 239–248, March 2005.

[30] P. Riba, J. Lladãs, and A. Fornés, “Handwritten word spotting by inexact matching of grapheme graphs,” in *Proc. of International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2015, pp. 781–785.

[31] A. Shamaie and A. Sutherland, “Graph-based matching of occluded hand gestures,” in *Proc. of Applied Imagery Pattern Recognition Workshop,*, Oct 2001, pp. 67–73.

[32] M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in *Proc. of International Conference on Computer Vision (ICCV) Volume 1*, vol. 2, Oct 2005, pp. 1482–1489.

[33] W. Goh, “Strategies for shape matching using skeletons,” *Computer vision and image understanding*, vol. 110, no. 3, pp. 326–345, 2008.

[34] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, “Discrete signal processing on graphs: Sampling theory,” *in IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6510–6523, 2015.

[35] 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.

[36] M. Trifan, B. Ionescu, C. Gadea, and D. Ionescu, “A graph digital signal processing method for semantic analysis,” in *International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 2015, pp. 187–192.

[37] H. Nguyen and M. Do, “Downsampling of signals on graphs via maximum spanning trees,” *in IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 182–191, 2015.

[38] S. Hemalatha and P. Valsalal, “Identification of optimal path in power system network using Bellman Ford algorithm,” *Modelling and Simulation in Engineering*, vol. 2012, p. 28, 2012.

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

[40] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[41] S. Zhong, Y. Hu, and J. Lu, “A new geometric-transformation robust and practical embedding scheme for watermarking 2D vector maps in the graph spectral domain,” in *Proc. of International Conference on Communications, Circuits and Systems*, vol. 1. IEEE, 2006, pp. 24–30.

[42] M. Fiedler, “A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory,” *Czechoslovak Mathematical Journal*, vol. 25, no. 4, pp. 619–633, 1975.

[43] M. Saerens, F. Fouss, L. Yen, and P. Dupont, “The principal components analysis of a graph, and its relationships to spectral clustering,” in *Proc. of European Conference on Machine Learning*. Springer, 2004, pp. 371–383.

[44] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian,” *in IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, Jan 2006.

[45] F. Perazzi, O. Sorkine-Hornung, and A. Sorkine-Hornung, “Efficient salient foreground detection for images and video using Fiedler vectors.” in *Proc. of WICED*. Citeseer, 2015, pp. 21–29.

[46] A. Sandryhaila and J. Moura, “Discrete signal processing on graphs: Frequency analysis,” *in IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[47] D. I. Shuman, M. J. Faraji, and P. Vandergheynst, “A multiscale pyramid transform for graph signals,” *in IEEE Transactions on Signal Processing*, vol. 64, no. 8, pp. 2119–2134, 2015.

[48] S. K. Narang and A. Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” *in IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2786–2799, June 2012.

[49] R. Coifman and M. Maggioni, “Diffusion wavelets,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53–94, 2006.

[50] S. K. Narang, Y. Chao, and A. Ortega, “Graph-wavelet filterbanks for edge-aware image processing,” in *Statistical Signal Processing Workshop (SSP)*. IEEE, 2012, pp. 141–144.

[51] N. Leonardi and D. Van De Ville, “Tight wavelet frames on multislice graphs,” *in IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3357–3367, 2013.

[52] M. Crovella and E. Kolaczyk, “Graph wavelets for spatial traffic analysis,” in *Proc. of Computer and Communications Societies Conference,*, vol. 3. IEEE, 2003, pp. 1848–1857.

[53] F. A. Huang, C. Y. Su, and T. T. Chu, “Kinect-based mid-air handwritten digit recognition using multiple segments and scaled coding,” in *Proc. of International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, Nov 2013, pp. 694–697.

[54] C. Wang, C. Y. Su, and C. L. Lin, “A novel recognition system for digits writing in the air using coordinated path ordering,” in *Proc. of International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Nov 2015, pp. 244–249.

[55] T. T. Chu and C. Y. Su, “A Kinect-based handwritten digit recognition for TV remote controller,” in *Proc. of International Symposium onIntelligent Signal Processing and Communications Systems (ISPACS)*, Nov 2012, pp. 414–419.

[56] J. S. Sheu and Y. L. Huang, “Implementation of an interactive TV interface via gesture and handwritten numeral recognition,” *Multimedia Tools and Applications*, pp. 1–22, 2015.

[57] T. Murata and J. Shin, “Hand gesture and character recognition based on Kinect sensor,” *International Journal of Distributed Sensor Networks*, vol. 1, pp. 1–6, July 2014.

[58] O. Mendels, H. Stern, and S. Berman, “User identification for home entertainment based on free-air hand motion signatures,” *in IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 11, pp. 1461–1473, 2014.

[59] M. Chen, G. Alregib, and B. H. Juang, “Air-writing recognition part I: Modeling and recognition of characters, words, and connecting motions,” *in IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 403–413, 2016.

[60] D. Frolova, H. Stern, and S. Berman, “Most probable longest common subsequence for recognition of gesture character input,” *in IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 871–880, 2013.

[61] B. D. de Vos, J. M. Wolterink, P. A. de Jong, M. A. Viergever, and I. Išgum, “2D image classification for 3D anatomy localization: employing deep convolutional neural networks,” in *Proc of Medical Imaging 2016: Image Processing*, vol. 9784. International Society for Optics and Photonics, 2016, p. 97841Y.

[62] 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.

[63] J. Li, B. Chen, and G. Lee, “SO-Net: Self-organizing network for point cloud analysis,” in *Proc. of Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9397–9406.

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

[65] D. Maturana and S. Scherer, “Voxnet: A 3D convolutional neural network for real-time object recognition,” in *Proc. of international conference on Intelligent Robots and Systems (IROS),*. IEEE, 2015, pp. 922–928.

[66] 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.

[67] 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.

[68] 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.

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

[70] 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.

[71] B. Wang and Y. Gao, "Structure integral transform versus Radon transform: A 2D mathematical tool for invariant shape recognition," *in IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5635–5648, 2016.

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

[73] H. Chiu, L. Kaelbling, and T. Lozano-Pérez, "Virtual training for multi-view object class recognition," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[74] S. S. Farfade, M. J. Saberian, and L. J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proc. of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 643–650.

[75] 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.

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

[77] 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.

[78] 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.

[79] 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.

[80] Y. Zhang, F. Jiang, S. Rho, S. Liu, D. Zhao, and R. Ji, "3D object retrieval with multi-feature collaboration and bipartite graph matching," *Neurocomputing*, vol. 195, pp. 40–49, 2016.

[81] A. Bernstein and C. Stein, "Fully dynamic matching in bipartite graphs," in *Proc. of International Colloquium on Automata, Languages, and Programming*. Springer, 2015, pp. 167–179.

[82] X. Zhu, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph PCA hashing for similarity search," *in IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2033–2044, 2017.

[83] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[84] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[85] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," in *Proc. of Shape Modeling Applications*. IEEE, 2004, pp. 145–156.

[86] J. Yan, X. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," in *Proc. of the ACM on International Conference on Multimedia Retrieval*. ACM, 2016, pp. 167–174.

[87] M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl, “3D shape histograms for similarity search and classification in spatial databases,” in *Symposium on Spatial Databases*. Springer, 1999, pp. 207–226.

[88] 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.

[89] F. Liu, D. Zhang, and L. Shen, “Study on novel curvature features for 3D fingerprint recognition,” *Neurocomputing*, vol. 168, pp. 599–608, 2015.

[90] Z. Lian, A. Godil, T. Fabry, T. Furuya, J. Hermans, R. Ohbuchi, C. Shu, D. Smeets, P. Suetens, and D. Vandermeulen, “Shrec’10 track: Non-rigid 3D shape retrieval.” *3DOR*, vol. 10, pp. 101–108, 2010.

[91] M. Abboud, A. Benzinou, K. Nasreddine, and M. Jazar, “Robust statistical shape analysis based on the tangent shape space,” in *Proc. of Image Processing (ICIP)*. IEEE, 2015, pp. 3520–3524.

[92] 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.

[93] 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.

[94] D. Vranic and D. Saupe, “Description of 3D-shape using a complex function on the sphere,” in *Proc. of International Conference on Multimedia and Expo*, vol. 1. IEEE, 2002, pp. 177–180.

[95] H. Cheng, L. Yang, and Z. Liu, “A survey on 3D hand gesture recognition,” *in IEEE Transactions on Circuits and Systems for Video Technology*, 2015.

[96] P. K. Pisharady and M. Saerbeck, “Recent methods and databases in vision-based hand gesture recognition: A review,” *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015.

[97] M. Asad and C. Abhayaratne, “Kinect depth stream pre-processing for hand gesture recognition,” in *Proc. of International Conference on Image Processing*. IEEE, Sept 2013, pp. 3735–3739.

[98] G. Plouffe and A. M. Cretu, “Static and dynamic hand gesture recognition in depth data using dynamic time warping,” *in IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 2, pp. 305–316, Feb 2016.

[99] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, “Robust part-based hand gesture recognition using Kinect sensor,” *in IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, Aug 2013.

[100] I. Oikonomidis, N. Kyriazis, and A. Argyros, “Efficient model-based 3D tracking of hand articulations using Kinect.” in *Proc. of British Machine Vision Conference (BMVC)*, vol. 1, no. 2, 2011, p. 3.

[101] M. de La Gorce, D. J. Fleet, and N. Paragios, “Model-based 3D hand pose estimation from monocular video,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1793–1805, 2011.

[102] L. Minto and P. Zanuttigh, “Exploiting silhouette descriptors and synthetic data for hand gesture recognition,” *Citeseer*, 2015.

[103] J. Triesch and C. von der Malsburg, “Robust classification of hand postures against complex backgrounds,” in *Proc. of International Conference on Automatic Face and Gesture Recognition*, Oct 1996, pp. 170–175.

[104] P. P. Kumar, P. Vadakkepat, and L. A. Poh, “Graph matching based hand posture recognition using neuro-biologically inspired features,” in *Proc. of International Conference on Control Automation Robotics Vision (ICARCV)*, Dec 2010, pp. 1151–1156.

[105] Y. T. Li and J. P. Wachs, “Recognizing hand gestures using the weighted elastic graph matching (WEGM) method,” *Image and Vision Computing*, vol. 31, no. 9, pp. 649–657, 2013.

[106] ——, “HEGM: A hierarchical elastic graph matching for hand gesture recognition,” *Pattern Recognition*, vol. 47, no. 1, pp. 80–88, 2014.

[107] H. Hamer, K. Schindler, E. Koller-Meier, and L. V. Gool, “Tracking a hand manipulating an object,” in *Proc. of International Conference on Computer Vision*, Sept 2009, pp. 1475–1482.

[108] M. Avraam, “Static gesture recognition combining graph and appearance features,” *International Journal of Advanced Research in Artificial Intelligence (IJARAI)*, vol. 3, no. 2, 2014.

[109] V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual interpretation of hand gestures for human-computer interaction: a review,” *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, Jul 1997.

[110] M. A. Aowal, A. S. Zaman, S. M. Rahman, and D. Hatzinakos, “Static hand gesture recognition using discriminative 2D Zernike moments,” in *Proc. of TENCON Region 10*. IEEE, 2014, pp. 1–5.

[111] K. C. Otiniano-Rodriguez, G. Camara-Chavez, and D. Menotti, “Hu and Zernike moments for sign language recognition,” in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV).* The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012, p. 1.

[112] N. H. Dardas and N. D. Georganas, “Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques,” *in IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[113] C. W. Ng and S. Ranganath, “Real-time gesture recognition system and application,” *Image and Vision computing*, vol. 20, no. 13, pp. 993–1007, 2002.

[114] D. Tang, J. Taylor, P. Kohli, C. Keskin, and J. Shotton, “Opening the black box: Hierarchical sampling optimization for estimating human hand pose,” in *Proc. of the IEEE International Conference on Computer Vision*, 2015, pp. 3325–3333.

[115] B. Feng, F. He, X. Wang, Y. Wu, H. Wang, S. Yi, and W. Liu, “Depth-projection-map-based bag of contour fragments for robust hand gesture recognition,” *in IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 511–523, 2017.

[116] C. Wang, Z. Liu, and S. C. Chan, “Superpixel-based hand gesture recognition with Kinect depth camera,” *in IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 29–39, 2015.

[117] Z. H. Chen, J. T. Kim, J. Liang, J. Zhang, and Y. B. Yuan, "Real-time hand gesture recognition using finger segmentation," *The Scientific World Journal*, vol. 2014, 2014.

[118] S. Hussain, R. Saxena, X. Han, J. Khan, and H. Shin, "Hand gesture recognition using deep learning," in *Proc. of International SoC Design Conference (ISOCC)*. IEEE, 2017, pp. 48–49.

[119] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. of Conference on computer vision and pattern recognition workshops*. IEEE, 2015, pp. 1–7.

[120] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network," in *Proc. of Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 4207–4215.

[121] A. A. Alani, G. Cosma, A. Taherkhani, and T. McGinnity, "Hand gesture recognition using an adapted convolutional neural network with data augmentation," in *Proc. of International Conference on Information Management (ICIM)*. IEEE, 2018, pp. 5–12.

[122] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *in ACM Transactions on graphics (tog)*, vol. 33, no. 5, p. 169, 2014.

[123] H. Liang, J. Yuan, and D. Thalmann, "Parsing the hand in depth images," *in IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1241–1253, 2014.

[124] M. Chen, G. AlRegib, and B. H. Juang, "Feature processing and modeling for 6D motion gesture recognition," *in IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 561–571, 2013.

[125] D. Fernández-Mota, J. Lladós, and A. Fornés, "A graph-based approach for segmenting touching lines in historical handwritten documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 3, pp. 293–312, 2014.

[126] B. Alwaely and C. Abhayaratne, "Graph spectral domain feature representation for in-air drawn number recognition," in *Proc. of European Signal Processing Conference (EUSIPCO)*, 2017, pp. 370–374. [Online]. Available: https://www.youtube.com/watch?v=cjY7UKTI-i4

[127] ——, "In-air hand-drawn numbers and shapes dataset," November 2018. [Online]. Available: https://figshare.shef.ac.uk/articles/In-Air_Hand-Drawn_Number_and_Shape_Dataset/7381472/1

[128] K. Khoshelham, "Accuracy analysis of Kinect depth data," in *Proc. of laser scanning workshop ISPRS*, vol. 38, no. 5, 2011, p. W12.

[129] S. Butler, "Algebraic aspects of the normalized Laplacian," in *Recent Trends in Combinatorics*. Springer, 2016, pp. 295–315.

[130] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[131] 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.

[132] 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.

[133] 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.

[134] 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.

[135] 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.

[136] 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.

[137] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman, “Deep convolutional networks do not classify based on global object shape,” *PLoS computational biology*, vol. 14, no. 12, p. e1006613, 2018.

[138] C. E. Connor, S. L. Brincat, and A. Pasupathy, “Transformation of shape information in the ventral pathway,” *Current opinion in neurobiology*, vol. 17, no. 2, pp. 140–147, 2007.

[139] S. L. Brincat and C. E. Connor, “Underlying principles of visual shape selectivity in posterior inferotemporal cortex,” *Nature neuroscience*, vol. 7, no. 8, p. 880, 2004.

[140] T. Martinetz and K. Schulten, “A neural-gas network learns topologies,” *Artificial Neural Networks*, pp. 397–402, 1991. [Online]. Available: https://goo.gl/kxKTLs

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

[142] B. Alwaely and C. Abhayaratne, “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. [Online]. Available: http://charith-abhayaratne.staff.shef.ac.uk/rdicassp19gs3d.html

[143] M. Akimaliev and M. Demirci, “Improving skeletal shape abstraction using multiple optimal solutions,” *Pattern Recognition*, vol. 48, no. 11, pp. 3504–3515, 2015. [Online]. Available: https://goo.gl/JadMMb.

[144] 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. [Online]. Available: https://goo.gl/iqpYVa.

[145] 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. [Online]. Available: https://goo.gl/TUe3Z2.

[146] 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. [Online]. Available: https://goo.gl/TUe3Z2

[147] 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. [Online]. Available: https://goo.gl/983ZTn

[148] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *in ACM Transactions on graphics (tog)*, vol. 28, p. 73, 2009. [Online]. Available: https://goo.gl/Dx2WTs

[149] C. L. Hwang and K. Yoon, "Methods for multiple attribute decision making," in *Multiple attribute decision making*. Springer, 1981, pp. 58–191.

[150] M. A. Khabou, L. Hermi, and M. H. Rhouma, "Shape recognition using eigenvalues of the Dirichlet Laplacian," *Pattern recognition*, vol. 40, no. 1, pp. 141–153, 2007.

[151] 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.

[152] 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.

[153] 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.

[154] H. Qiu and E. R. Hancock, "Graph matching and clustering using spectral partitions," *Pattern Recognition*, vol. 39, no. 1, pp. 22–34, 2006.

[155] A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2D static hand gesture colour image dataset for ASL

gestures," *Research Letters in the Information and Mathematical Sciences*, vol. 15, pp. 12–20, 2011. [Online]. Available: http://www.massey.ac.nz/massey/fms/Colleges/College%20of%20Sciences/IIMS/RLIMS/Volume%2015/GestureDatasetRLIMS2011.pdf

[156] A. Agathos, I. Pratikakis, P. Papadakis, S. Perantonis, P. N. Azariadis, and N. S. Sapidis, "Retrieval of 3D articulated objects using a graph-based representation." *3DOR*, vol. 2009, pp. 29–36, 2009.

[157] G. Lavoué, "Combination of bag-of-words descriptors for robust partial shape retrieval," *The Visual Computer*, vol. 28, no. 9, pp. 931–942, 2012.

[158] M. Ye, V. Stankovic, L. Stankovic, and G. Cheung, "Deep graph regularized learning for binary classification," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 3537–3541.