UNIVERSITÀ CA' FOSCARI VENEZIA

DOTTORATO DI RICERCA IN INFORMATICA, 26° CICLO

(A.A. 2010/2011 – 2012/2013)

# Modeling, Classification and Analysis of Graph Structures

SETTORE SCIENTIFICO DISCIPLINARE DI AFFERENZA: INF/01

TESI DI DOTTORATO DI LUCA ROSSI
MATR. 955857

TUTORE DEL DOTTORANDO
Andrea Torsello

COORDINATORE DEL DOTTORATO
Riccardo Focardi

August, 2013

Author's Web Page: `http://www.dsi.unive.it/~lurossi`

Author's e-mail: lurossi@dsi.unive.it

Author's address:

Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348465
fax. +39 041 2348419
web: `http://www.dais.unive.it`

This thesis is dedicated to my beloved family and 老婆. Thank you for all your love, support and constant encouragement.

# Abstract

Graph-based representations have become increasingly popular due to their ability to characterize in a natural way a large number of systems which are best described in terms of their structure. Concrete examples include the use of graphs to represent shapes, metabolic networks, protein interactions, scientific collaboration networks and road maps. The challenge in this domain is that of developing efficient tools for the analysis and classification of graphs, a task which is far from being trivial due to their rich expressiveness.

In this thesis, we introduce a novel algorithm to extract an undirected graph from a 3d shape. Then, we show how to learn a graph generative model which is able to capture the modes of structural variation within a set of graphs, and how to select the optimal model using a classical MML approach, as well as a novel information-theoretic framework. Shifting our focus from generative to deterministic classification approaches, we then introduce a family of graph kernels, which are based on a quantum-mechanical analysis of the structure of the graph. This leads us to the final part of the thesis, where we propose a series of algorithms for the analysis of graph structure which build on the common idea of exploiting the correlation between structural symmetries and the interference properties of quantum walks.

# Sommario

Le rappresentazioni basate su grafi hanno visto crescere negli anni la loro popolarità per la loro capacità di descrivere in maniera naturale un ampio numero di sistemi caratterizzati da una forte componente strutturale. Esempi concreti includono l'uso di grafi per rappresentare forme, reti metaboliche, interazioni proteiche, collaborazioni scientifiche e mappe stradali. In questo campo, la sfida è quella di sviluppare strumenti efficienti per l'analisi e la classificazione di grafi, un compito reso ancora più arduo dalle ricche potenzialità espressive dei grafi.

La tesi comincia con l'introduzione di un nuovo algoritmo per l'estrazione di un grafo indiretto da un oggetto tridimensionale. Mostriamo quindi come imparare un modello generativo in grado di catturare le variazioni strutturali all'interno di un insieme di grafi, e come selezionare il modello ottimale usando un classico approccio MML, oltre ad un nuovo sistema basato sulla teoria dell'informazione. Spostiamo a questo punto l'attenzione dagli algoritmi di classificazione generativi a quelli deterministici, ed in questo contesto introduciamo una nuova famiglia di kernel su grafi che si basano su un'analisi meccanico-quantistica delle struttura dei grafi. Questo ci porta alla parte finale della tesi, dove proponiamo una serie di algoritmi per l'analisi della struttura dei grafi che si basano sull'idea comune di sfruttare la correlazione tra simmetrie strutturali e i pattern di interferenza tipici dei cammini quantistici.

# Acknowledgments

There is little doubt that I have been blessed with possibly the best academic genealogy I could ask for. Some thanks are in order here, and I will start from the root: my academic grandfather, Prof. Edwin R. Hancock. I feel extremely honoured to have been supervised by Edwin Hancock during my research visit to York. There is no doubt that thanks to his insights I managed to get the best out of my academic research. Not to mention the fact that, partly because of his fondness for a specific Chinese restaurant in York, I also happened to fall in love with Chinese (food).

And now, to my academic father. I would like to express my deepest gratitude to my supervisor Dr. Andrea Torsello for his continuous support during my PhD study, his patience and for making me a better (and hopefully smarter) scientist and person. I regret to admit it, but I feel that, despite many years spent under his guidance, what I was able to learn from him isn't but a tiny percentage of his knowledge.

Finally, I wish to thank Prof. Francisco Escolano and Prof. Simone Severini for their assessment of this thesis and their valuable feedback.

# Contents

# List of Figures

# List of Tables

# Preface

This thesis covers the research which I carried out during my three years of PhD at the Department of Environmental Sciences, Informatics and Statistics of Ca' Foscari University of Venice, Italy. Although I started by improving and extending my MSc thesis on medial surfaces and graph representations of shapes, I soon realized that the big challenge actually lied in the so-called field of graph-based pattern recognition. I consider myself extremely lucky for having had the chance of researching on such a diverse and multi-faceted topic, as this led me to develop a broad spectrum of expertise which in my opinion is essential for conducting academic research at the highest level possible. I should mention, finally, that the six months that I spent working at the University of York, UK, had a key role in the shaping of my scientific and technical knowledge. Unfortunately, the continuous pressure in academia to rapidly and continuously publish novel work too often leads to scarcer collaborations among different research groups, as in the race for novelty there is no room for second place. But the sharing of ideas lies at the very heart of scientific research, and only by discussing and sharing our ideas within the community our work can really bloom.

Before overviewing the contents of this thesis in the next Chapter, I would like to end with a quote that best illustrates what should always remain the true driving force of academic research, and what has actually pushed me to become a PhD student: the passion for science.

> The feeling of awed wonder that science can give us is one of the highest experiences of which the human psyche is capable. It is a deep aesthetic passion to rank with the finest that music and poetry can deliver. It is truly one of the things that make life worth living and it does so, if anything, more effectively if it convinces us that the time we have for living is quite finite.
>
> Richard Dawkins, Unweaving the Rainbow: Science, Delusion and the Appetite for Wonder.

# Published Papers

[1] Albarelli, A., Bergamasco, F., Rossi, L., Vascon, S., and Torsello, A. A stable graph-based representation for object recognition through high-order matching. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, IEEE, pp. 3341–3344, 2012.

[2] Bai, L., Hancock, E. R., Torsello, A., and Rossi, L. A quantum jensen-shannon graph kernel using the continuous-time quantum walk. In *Graph-Based Representations in Pattern Recognition*. Springer Berlin Heidelberg, pp. 121–131, 2013.

[3] Han, L., Rossi, L., Torsello, A., Wilson, R. C., and Hancock, E. R. Information theoretic prototype selection for unattributed graphs. In *Structural, Syntactic, and Statistical Pattern Recognition*. Springer Berlin Heidelberg, pp. 33–41, 2012.

[4] Rossi, L., and Torsello, A. Coarse-to-fine skeleton extraction for high resolution 3d meshes. *Computer Vision and Image Understanding, Accepted for Publication.*

[5] Rossi, L., and Torsello, A. An adaptive hierarchical approach to the extraction of high resolution medial surfaces. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, IEEE, pp. 371–378, 2012.

[6] Rossi, L., Torsello, A., and Hancock, E. R. Approximate axial symmetries from continuous time quantum walks. In *Structural, Syntactic, and Statistical Pattern Recognition*, Springer Berlin Heidelberg, pp. 144–152, 2012.

[7] Rossi, L., Torsello, A., and Hancock, E. R. A continuous-time quantum walk kernel for unattributed graphs. In *Graph-Based Representations in Pattern Recognition*, Springer Berlin Heidelberg, pp. 101–110, 2013.

[8] Rossi, L., Torsello, A., Hancock E. R., and Wilson, R. Characterising graph symmetries through quantum-jensen shannon divergence. Physical Review E, 88(3), 032806, 2013.

[9] Rossi, L., Torsello, A., and Hancock, E. R. Attributed graph similarity from the quantum jensen-shannon divergence. In *2nd International Workshop on Similarity-Based Pattern Analysis and Recognition*, Springer Berlin Heidelberg, pp. 204–218, 2013.

[10] ROSSI, L., TORSELLO, A., AND HANCOCK, E. R. Enhancing the quantum jensen-shannon divergence kernel through manifold learning. In *15th International Conference on Computer Analysis of Images and Patterns*, Springer Berlin Heidelberg, pp. 62–69, 2013.

[11] TORSELLO, A., AND ROSSI, L. Supervised learning of graph structure. In *Similarity-Based Pattern Recognition*, Springer Berlin Heidelberg, pp. 117–132, 2011.

[12] BAI, L., ROSSI, L., TORSELLO, A., AND HANCOCK, E. R. A Quantum Jensen-Shannon Graph Kernel From The Continuous-Time Quantum Walk. *Pattern Recognition, Submitted (08/2013).*

# 1
# Introduction

The focus of this thesis is on the mathematical object known as *graph*. Historically, the theory of graphs has its roots back in 1736, when Euler was trying to solve the problem of finding a walk through the city of Königsberg such that each bridge would have been crossed exactly once. As this small yet important example suggests, the fortune of graphs lies in their ability to intuitively model a vast variety of real-world problems. In particular, graph-based representations have long been used in computer science to characterize in a natural way a large number of objects which are best described in terms of their structure, with applications in fields as diverse as computer vision, bioinformatic, linguistic and sociology. However, the rich expressiveness of graphs usually comes at the cost of an increased difficulty in applying standard pattern recognition techniques on them. The challenge, in this sense, is that of developing efficient tools for the analysis and classification of graphs, a task which proved to be far from trivial.

This thesis attempts to address these issues by introducing a wide spectrum of novel techniques for the modeling, classification and analysis of graph structures. Chapter 3 introduces a novel algorithm for the extraction of medial surfaces from three-dimensional shapes, where the medial surface can be seen as an intermediate representation in the process of extracting a graph from a shape. Chapter 4 and Chapter 5 cover the problem of graph classification using generative and discriminative approaches, respectively. Finally, in Chapter 6 we shift our attention to the problem of analyzing the structure of a graph.

## Graph-Based Representations: Medial Surfaces

Graph-based representations have long been used to model 2D and 3D shapes in computer vision applications, in an attempt to automatically analyze and classify objects. Today, the wide availability of cheap 3D scanning devices renders topical the automated extraction of a representation which provides a simple venue to perform shape analysis and representation under deformation and articulation. For this reason, the design of efficient algorithms for 3D skeleton extraction is of pivotal importance. The 3D skeleton, or medial surface, of an object, is usually defined as the locus of the centres of the maximal inscribed spheres which are at least bitangent to the shape boundary. While in 2D the image needs to be segmented in order to extract the skeleton, in 3D the objects are naturally modeled as distinct meshes, thus rendering the skeletonization much more practical. The addition of a third dimension, however, renders the task of medial surfaces extraction particularly challenging. First, there is an exponential growth of the number of voxels, which may render the computation impracticable when a high resolution is needed. Further, volumetric objects are commonly represented as triangle meshes, that may eventually need to be voxelized before any further computation is done. Depending on the resolution chosen, this discretization might yield the wrong topology. Moreover, not only the spatial and time complexity of the algorithm is increased, but also tasks that are almost trivial in two dimensions, such as ensuring the topological equivalence between the object and its skeleton, become more challenging when a third dimension is added.

We deal with the problem of medial surfaces extraction in Chapter 3, where we propose a hierarchical algorithm where we iteratively decide whether if refining a voxel or not based on the local value of the divergence of the momentum field, i.e., the confidence that we have in a point being skeletal. With the medial surface to hand, it is possible to represent the relations between its parts as an unattributed graph, which then may provide the starting point for a shape recognition pipeline.

## Graph-Based Pattern Recognition

Standard classification techniques can be usually divided into two broad categories, namely the *generative* and the *discriminative* approaches. Let $x$ be a data belonging to class $y$. The generative approach will try estimate the joint probability density function $p(x, y)$, or, equivalently, $p(x|y)$ and $p(y)$. The classification is then performed using $p(y|x)$, which is obtained by applying Bayes rule. On the other hand, discriminative approaches will try to estimate $p(y|x)$ directly from the data, which is equivalent to learn a classification function $y = f(x)$.

Unfortunately, our ability to analyze graphs is severely limited by the lack of explicit correspondences between the nodes of different graphs, and the variability of the graphs structure. This is clearly a problem for generative models, which in order to estimate $p(x|y)$ usually require the mapping of the nodes and edges of the observed graphs to the nodes and edges of the model graph. The problem is even worse in the case of discriminative approaches, which usually require the graphs to be first embedded into a vectorial space, a procedure which is far from being trivial. Again, the reason for this is that there is no canonical ordering for the nodes in a graph and a correspondence order must be established before analysis can commence. Moreover, even if a correspondence order can be established, graphs do not necessarily map to vectors of fixed length, as the number of nodes and edges can vary.

## Generative Approaches: Learning the Structure

Chapter 4 considers the problem of learning a generative graph model $\mathscr{G}$ that can be used to describe the distribution of structural data and characterize the structural variations present in the observed set $S = (g_1, \ldots, g_l)$. The proposed graph generative model works by decoupling the structural and stochastic parts and making the naïve assumption that the observation of each node and each edge is independent of the others, but allowing correlations to pop up by mixing the models. Moreover, we enhance the generalization capabilities of the approach by adding to the generative model the ability to add nodes which are not part of the core structure, thus not requiring to model explicitly isotropic random noise.

Here we deal with the problem of the hidden correspondences between the models and the observations by marginalizing the observation probability of a graph over all the possible correspondences. On the other hand, standard approaches, which assume the maximum likelihood estimation for the correspondences, or simply a single estimation, yield a bias in the estimation of the probability density function $p(x|y)$. However, averaging over all possible correspondences is clearly not possible due to the super-exponential growth of the set. Hence, in Section 4.1.1 we show how to solve the problem by resorting to an importance sampling estimation approach.

## Discriminative Approaches: Graph Kernels

The interest in generative approaches comes from their ability to better characterize the structural variation of the set. However, discriminative approaches usually show a higher classification performance. Unfortunately, standard discriminative techniques usually work on vectorial spaces, and thus we are once again confronted with the problem of establishing a correspondence between graphs and vectors. Kernel methods, whose best known example is furnished by support vector machines, provide a neat way to shift the problem from that of finding an embedding to that of defining a positive semidefinite kernel, via the well-known kernel trick. More precisely, rather then explicitly introducing a vectorial space, one needs to define the kernel measure between two graphs, which, if certain conditions are satisfied, will correspond to a dot product in an implicitly defined vectorial space. The literature is rich of successful attempts to define kernel between graphs. Most of these usually fall within the family of R-convolution kernels. The fundamental idea behind R-convolution kernels is that of defining a kernel between two discrete objects by decomposing them and comparing these simpler substructures.

Chapter 5 introduces a novel graph kernel which evaluates the similarity between two graphs through the evolution of a suitably defined continuous-time quantum walk on their structure. Quantum walks on graphs represent the quantum mechanical analogue of the classical random walk on a graph. Recently, there has been an increasing interest in using quantum walks as a primitive for designing novel quantum algorithms, as quantum walks have shown to possess unique properties which can lead to exponential speedups over their classical counterparts. These properties seem to be intimately related to the constructive and destructive interference effects of quantum processes, which are themselves tightly connected to the presence of symmetrical structures in the graph. In order to exploit this connection, given two graphs, we first merge them into a new structure whose degree of symmetry will be maximum when the original graphs are isomorphic. With this new graph to hand, we compute the density operators of the quantum systems representing the evolution of two suitably defined quantum walks. Finally, we define the kernel between the two original graphs as the quantum Jensen-Shannon divergence between these two density operators. Moreover, Section 5.5 shows how to further increase the performance of this kernel in a classification task. This is achieved by applying standard manifold learning techniques on the kernel embedding to map the data onto a low-dimensional space where the different classes can exhibit a better linear separation.

## Graph Structure Analysis

In Chapter 6 we move the focus from traditional graph-based pattern recognition techniques to the analysis of graph structure, in the more general framework of complex network science. Complex network science is rapidly gaining popularity among researchers due to its key role in understanding the massive amount of data which are produced every day by human activities, the most intuitive example being online social networks. More generally, complex network science is interested in the study of the properties of a large number of complex systems which are modeled as graphs, and are not limited to online social networks. A non-exhaustive list of examples includes metabolic networks, protein interactions, brain networks, vascular systems, scientific collaboration networks and road maps. Properties such as small-worldness and the power-law distribution of vertex degrees have been observed in these networks, suggesting a marked difference with Erdös-Rényi random graphs.

However, in recent years there has also been a growing interest in characterizing the presence of symmetries in real-world networks, which is in turn linked to the redundancy and robustness of networks. Inspired by the quantum mechanical analysis of Chapter 5, we propose again to exploit the correlation between structural symmetries and the interference properties of quantum walks. More precisely, Section 6.1 deals with the measurement of the degree of symmetry of a graph. Section 6.2, on the other hand, introduces a novel algorithm for the explicit detection of approximate axial symmetries where the graph structure is probed through the evolution of two suitably defined quantum walks. Finally, Section 6.3 is dedicated to the study of a novel measure of node centrality, i.e., the *importance* of a node in a network.

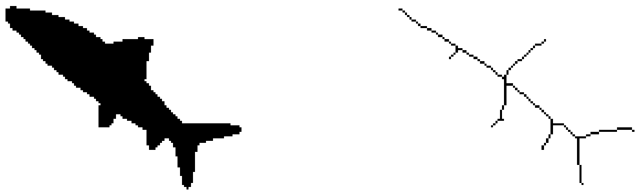
# 2 Related Work

This Chapter is intended to give a comprehensive overview of the existing literature covering the topics of this thesis. To this end, we will need to introduce a number of concepts which will be needed to understand the following Chapters. In particular, Section 2.4 will be dedicated to an overview of quantum computation and the relation between quantum walks and structural symmetries, as this idea will form the basis of the analysis of Chapter 5 and Chapter 6. Note, however, that none of the Sections of this Chapter is intended to provide an exhaustive survey of the state-of-art techniques in the relative topic. When needed, however, the reader will be provided with references to more rigorous surveys.

This Chapter is organized as follows. Section 2.1 reviews the main algorithms for 2D and 3D skeletons extractions. Section 2.2 illustrates the literature on graph generative models, while Section 2.3 provides an overview of graph kernels. In Section 2.4 we provide a review of the relevant literature on quantum computing, particularly the literature on the quantum walk, and we discuss the relation between quantum walks and structural symmetries. Finally, Section 2.5 illustrates some concepts and algorithms of complex network science, with particular emphasis on structural symmetries and centrality measures.

## 2.1 Medial Surfaces Extraction

The skeleton of a 2D shape, or medial axis transform, is defined as the locus of the centers of the maximal inscribed circles bitangent to the shape boundary. This shape descriptor has found a large number of successful applications in computer vision, due to a number of important properties. The skeleton is in fact a concise representation of the original shape and it is topologically equivalent to it. Moreover, the skeleton is invariant to several shape deformations, which include rotations, changes of viewpoint and changes of scale. The 3D equivalent of the skeleton is the medial surface, sometimes called 3D skeleton. Although it retains the interesting properties of its two-dimensional counterpart, the 3D skeleton poses a series of challenges which make it considerably harder to compute. A non exhaustive list of applications of skeletons includes 2-D and 3-D shape recognition [110, 129], volumetric models deformation [28, 150], segmentation [114] and protein structure identification [13].



(a) Original Shape (b) 2D Skeleton

Figure 2.1: A 2D shapes and the medial axis extracted with a thinning algorithm.

In the following Subsections we provide a review of the 2D and 3D skeleton extraction algorithms present in the literature. For a more exhaustive survey of the topic see [129].

### 2.1.1 2D Skeletons

Over the years several methods have been proposed to compute the 2D skeleton of a shape, but all of them can be can be basically divided into four main categories.

#### Thinning Methods

The first class of methods are the thinning ones, which simulate Blum's grassfire transform by iteratively eroding layers from the shape [26] [50]. During the thinning procedure care must be given not to change the object topology and to ensure the correct geometrical position of the skeleton with respect to the original shape, since the result is clearly dependent on the order in which the erosion is performed. Unfortunately, while fast and simple to implement, these algorithms are quite sensitive to Euclidean transformations, so they typically fail to locate accurately the skeleton of the object. Fig. 2.1 shows the example of a 2D shape and the medial axis extracted using a thinning method.

#### Distance Transform Based Methods

The second class of methods exploits the fact that the skeleton coincides with the local extrema of the Euclidean distance transform [41] [63] [87]. This in turn relies on the computation of the Euclidean distance between each point in the interior of the object and the boundary of the shape, which can be done in linear time $O(n)$, where $n$ is the number of pixel of the image [97]. These approaches then attempt to detect the ridges of the distance map either directly or by evolving a series of curves, such as snakes, under a potential energy field defined by the distance map. Although these methods fulfill the geometrical constraint, ensuring the topological correctness is not trivial.

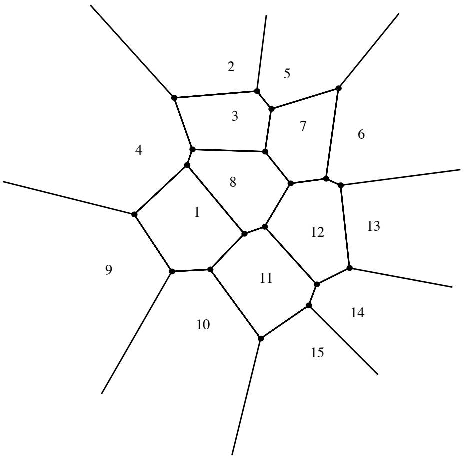


Figure 2.2: The Voronoi diagram of a set of two-dimensional points.

### Voronoi Diagram Methods

A third class of methods is based on the Voronoi diagram of a subset of the boundary points [106]. The Voronoi diagram of a set of points (called generating points) is defined as the partition of the space into regions so that each region contains the generating point *p* and the locus of the points that are nearer to *p* than to any other generating point. Fig. 2.2 shows the Voronoi diagram of a set of 15 two-dimensional points. The idea of these approaches is that, under appropriate smoothness conditions, the Voronoi diagram of a subset of the boundary points converges to the skeleton as the number of the sampled boundary points increases. Fig. 2.3 shows 16 points uniformly sampled along the boundary of a rectangle, and the resulting skeleton is highlighted in red.

These methods ensure topology preservation and invariance under Euclidean transformations, in addition to locate the skeleton with great accuracy, provided that the boundary of the shape is sampled densely enough. However, if the object being skeletonized is not a polygon, they obviously suffer from limitations due to the computational complexity of finding the Voronoi diagram of the shape (or alternatively the Delaunay triangulation). Moreover, approximating a smooth shape with many straight line segments introduces a lot of spurious branches, which then need to be pruned with techniques typically based on heuristics.

### Differential Methods

The fourth, and final, class of methods is based on the analysis of the differential structure of the boundary. In [82], the boundary is segmented at points of maximal curvature and the authors show that the skeleton is a subset of the Voronoi diagram of these segments. Despite its accuracy, the main drawback of this approach is the need to estimate the boundary curvature by fitting a curve to it, which is a computational

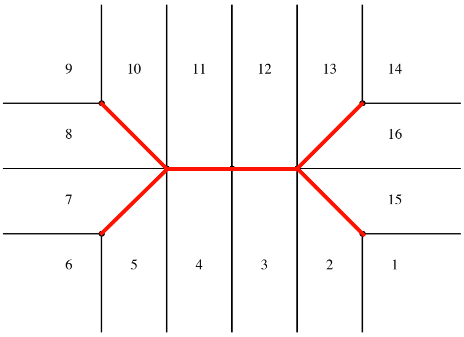


Figure 2.3: The skeleton is a subset of the Voronoi diagram.

demanding and quite delicate task. A somehow similar approach is that of Leymarie and Leving [87], which is based on the concept of active contours introduced in [78]. Kass, Witkin and Terzopoulos cast the problem of boundary location into a curve evolution framework, where the curve is evolved in a potential energy field under certain smoothness constraints. By using the distance map as the energy function, Leymarie and Leving are able to estimate the shape skeleton by simulating the grassfire transform and identifying the points where the wavefront collapses as the skeletal points. Unfortunately, as in [82] this requires an initial segmentation of the boundary at curvature extrema, which is itself a challenging problem.

Another important method that belongs to this class stems from the Hamiltonian analysis of the boundary flow dynamics [127]. Siddiqi et al. state that the singular points where the system ceases to be Hamiltonian (i.e., an energy conservation principle is violated) are responsible for the formation of skeletal points. Their analysis, however, is inevitably flawed by the fact that they don't take into account the effects of the boundary curvature, a problem which they only partially solve in [128]. Subsequently, however, Torsello and Hancock [138] show how to completely overcome the problem by performing a Hamilton-Jacobi analysis of the flow under conditions where the flow density varies due to curvature.

### 2.1.2 3D Skeletons

Similarly to its two-dimensional counterpart, the 3D skeleton (medial surface) of a volumetric object can be defined as the locus of the centers of the maximal inscribed spheres which are at least bitangent to the shape boundary. Note that in the literature there are two competing 3D generalizations of the skeleton: the curve (or line) skeleton [45, 17], which provides a minimal representation for shape analysis and recognition, and the medial surfaces, which, on the other hand, carry enough information to

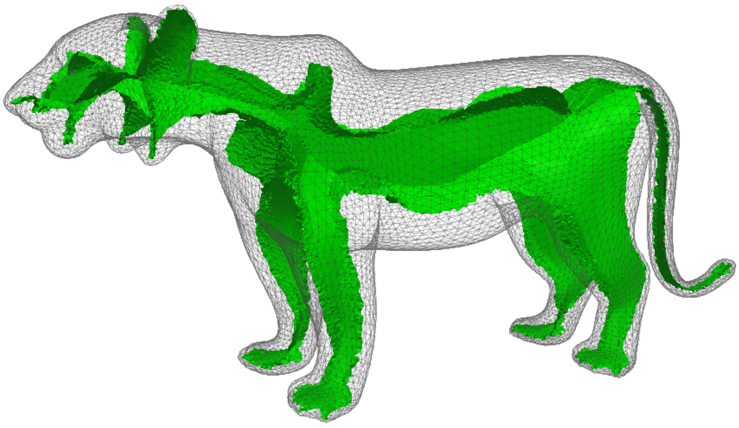


Figure 2.4: The medial surface is the locus of the centers of the maximal inscribed spheres which are at least bitangent to the shape boundary.

recover the original shape. Moreover the medial surface is topologically equivalent to the shape in the sense that there exists a homotety that maps its segments (considered as two oriented surfaces) to the original mesh. The same is not true of the line skeleton which is a lossy simplification of the shape. Finally, the curve skeleton is ill-defined in some degenerate cases, as for example the shape of a cup. In this thesis, we therefore concentrate on the extraction of medial surfaces from triangulated meshes.

Arcelli et al. [16] propose a distance-driven algorithm that is topology preserving but works only on voxelized objects, and thus is unable to cope with high resolution inputs. The same holds for the algorithm proposed by Siddiqi et al. [129], which is a generalization to three dimensions of the Hamilton-Jacobi skeleton and suffers from the same limitations of its two-dimensional counterpart, since it doesn't take into account the effects of boundary curvature. A more robust algorithm is that of Reniers et al. [115], where both the curve and the surface skeletons are located by means of an advection-based importance measure. Unfortunately this measure turns out to be well defined only for genus 0 shapes.

Another approach is that of Bai et al. [21] and Quadros et al. [111], who propose to use adaptive octrees in order to reduce the spatial and time complexity. This allows some parts to be discretized more densely while the rest can be analyzed at a coarser scale. However, both these approaches work on a precomputed octree, where the grid refinement criterion is based on simple heuristics. In [21] they propose to increase the grid resolution on those voxels that are roughly at the center of the shape, since the medial surface is supposed to be located approximately there. Anyway, they clearly state that the design of an optimal grid adaptation criterion for skeleton computation is beyond the scope of their paper, and a more efficient heuristic should be used instead. In [111] the octree nodes are generated according to the vertices and centroids of the facets of an input CAD model, therefore the density of the nodes is higher in the pres-

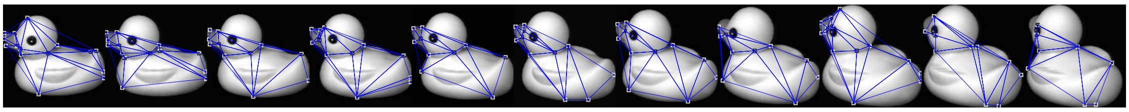


Figure 2.5: A set of increasingly rotated objects and the associated delaunay graphs representations. During the rotation, the set of interest points which undergo the delaunay triangulation changes, and as a result the structure of the graphs varies. Generative models can be used to help capturing the structural variations of the observed graphs.

ence of small features or regions of high curvature. The resulting skeleton, however, is disconnected, and it is composed of sets of nodes at different levels of resolution.

On the other hand Yoshizawa et al. [150] and Hisada et al. [71] take a Voronoi-based approach, where the skeleton of a mesh is approximated by a skeletal mesh having the same connectivity as the original mesh. The QuickHull algorithm [23] is used to extract the Voronoi diagram of the mesh vertices, then for each mesh vertex $v$ they define a skeletal point $p$ at a distance $d$ along $v$'s normal, where the displacement $d$ is computed as the distance from $v$ to the arithmetic mean of the Voronoi vertices of the Voronoi region containing $v$. The connectivity between skeletal vertices is then defined according to the connectivity between the corresponding mesh vertices. These approaches are fast and do not require an initial voxelization, but extract only an approximation of the skeleton and are extremely sensible to small perturbations of the boundary.

## 2.2 Graph Generative Models

Graph-based representations have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure, as they can concisely capture the relational arrangement of object primitives, in a manner which can be invariant to changes in object viewpoint. Despite their many advantages and attractive features, the methodology available for learning structural representations from sets of training examples is relatively limited, and the process of capturing the modes of structural variation for sets of graphs has proved to be elusive. Fig. 2.5 shows a set of 2D objects and the associated delaunay graph representations. As the object is rotated, the graph structure can change dramatically, thus making the ability of capturing this variation of pivotal importance.

Recently, there has been considerable interest in learning structural representations from samples of training data, in particular in the context of Bayesian networks, or general relational models [59]. The idea is to associate random variables with the nodes of the structure and to use a structural learning process to infer the stochastic dependency between these variables. However, these approaches rely on the availability of correspondence information for the nodes of the different structures used in learning. In many cases the identity of the nodes and their correspondences across samples of training data are not known, rather, the correspondences must be recovered from structure.

In the last few years, there has been some effort aimed at learning structural archetypes and clustering data abstracted in terms of graphs. In this context spectral approaches have provided simple and effective procedures. For example Luo and Hancock [109] use graph spectral features to embed graphs in a low dimensional space where standard vectorial analysis can be applied. While embedding approaches like this one preserve the structural information present, they do not provide a means of characterizing the modes of structural variation encountered and are limited by the stability of the graph's spectrum under structural perturbation.

Bonev et al. [30], and Bunke et al. [36] summarize the data by creating super-graph representation from the available samples, while White and Wilson [146] use a probabilistic model over the spectral decomposition of the graphs to produce a generative model of their structure. While these techniques provide a structural model of the samples, the way in which the supergraph is learned or estimated is largely heuristic in nature and is not rooted in a statistical learning framework. Torsello and Hancock [139] define a superstructure called tree-union that captures the relations and observation probabilities of all nodes of all the trees in the training set. The structure is obtained by merging the corresponding nodes and is critically dependent on the order in which trees are merged. Todorovic and Ahuja [134] applied the approach to object recognition based on a hierarchical segmentation of image patches and lifted the order dependence by repeating the merger procedure several times and picking the best model according to an entropic measure. However, the model structure and model parameter are tightly coupled, which forces the learning process to be approximated through a series of merges, and all the observed nodes must be explicitly represented in the model, which then must specify in the same way proper structural variations and random noise. The latter characteristic limits the generalization capabilities of the model.

An alternative to these approaches consists in computing the graph median or the generalized graph median [74, 56]. Given a set of observed graphs, the graph median is defined as the graph that minimizes the sum of the distances to all the observed graphs. The difference in the graph median and its generalized version lies in the fact that the former belongs to the set of observed graphs while the latter generally does not. Note that these kinds of approaches aim at learning a graph structural prototype, rather than a probabilistic model, where instead the goal is that of defining a probability distribution over the observed graphs.

Recently, Han et al. [66] introduced a method for learning a generative model of graphs which can be seen as an extension of [139]. The method is posed in terms of learning a supergraph from which the samples can be obtained by edit operations. After estimating the probability distributions for the occurrence of supergraph nodes and edges, the authors propose an EM approach to learn both the structure of the supergraph and the correspondences between the nodes of the observed graphs and those of the supergraph, which are treated as missing data. Similarly, Torsello [135] proposed a generalization for graphs which allowed to decouple structure and model parameters and used a stochastic process to marginalize the set of correspondences, however the approach does not deal with attributes and all the observed nodes still need be explicitly represented in the model. Further, the issue of model order selection was not addressed. Torsello and Dowe [136] addressed the generalization capabilities of the approach by adding to the generative model the ability to add nodes, thus not requiring to model explicitly isotropic random noise, however correspondence estimation in this approach was cumbersome and while it used a minimum message length principle for selecting model-complexity, that could be only used to choose from different learned structures since it had no way to change the complexity while learning the model.

Closely related to the problem of learning a model is that of selecting the optimal one from a set of candidate models. Standard model selection methods include the Minimum Message Length criterion (MML) [143], the Aikake [14] and the Bayesian Information Criteria [120]. Given a set of candidate models, one may be tempted to choose the one that best fits the training data as the optimal one, e.g., the one with the highest likelihood given the training data. However, this usually comes at the cost of overfitting the model to the observed data. As an example, consider the problem of estimating the best curve to fit 6 data points. A fifth-order polynomial can fix the points exactly, but it may be an overkill if, for example, the 6 points lie on a straight line. Moreover, if the points are affected by noise we would end up modeling the noise as well. For this reason, the optimality criterion should be a trade-off between the goodness of fit of the model and the complexity of the model. MML, AIC and BIC represent different ways to weight the goodness of fit of the model, i.e., its likelihood given the training data, and its complexity, usually expressed as a function of the number of parameters of the model.

## 2.3 Graph Kernels

Although generative approaches to graph classification seem attractive because of their ability of characterizing the modes of structural variation of graphs, discriminative approaches are known to provide a much better performance in terms of classification accuracy. Unfortunately, our ability to apply discriminative algorithms is severely limited by the restrictions posed by standard pattern recognition techniques, which usually require the graphs to be first embedded into a vectorial space, a procedure which is far from being trivial. The reason for this is that there is no canonical ordering for the nodes in a graph and a correspondence order must be established before analysis can commence. Moreover, even if a correspondence order can be established, graphs do not necessarily map to vectors of fixed length, as the number of nodes and edges can vary.

Kernel methods [119], whose best known example is furnished by support vector machines (SVMs) [141], provide a neat way to shift the problem from that of finding an embedding to that of defining a positive semidefinite kernel, via the well-known kernel trick. In fact, once we define a positive semidefinite kernel $k : X \times X \to \mathbb{R}$ on a set $X$, then we know that there exists a map $\phi : X \to H$ into a Hilbert space $H$, such that $k(x, y) = \phi(x)^{\top}\phi(y)$ for all $x, y \in X$. Thus, any algorithm that can be formulated in terms of scalar products of the $\phi(x)$s can be applied to a set of data on which we have defined our kernel. As a consequence, we are now faced with the problem of defining a positive semidefinite kernel on graphs rather than computing an embedding. However, due to the rich expressiveness of graphs, also this task has proven to be difficult.

Many different graph kernels have been proposed in the literature [44, 90, 60, 31, 125]. Graph kernels are generally instances of the family of R-convolution kernels introduced by Haussler [67]. The fundamental idea is that of defining a kernel between two discrete objects by decomposing them and comparing some simpler substructures. The initial attempts to define graph kernels actually focused on particular subsets or families of graphs, such as trees [44] or strings [90]. In [44] the authors introduce a kernel on trees which is used for natural language processing tasks, where the kernel function computes the number of common subtrees in two trees, while in [90], the kernel function computes the number of common subsequences between two string of characters. However, the limit of these kernels clearly lies in the fact that they only work on a small subset of graphs, thus losing the potential expressiveness of more general graphs.

To this hand, a number of more generic graph kernels have then been introduced in the literature. For example, Gärtner et al. [60] propose to count the number of common random walks between two graphs, while Borgwardt and Kriegel [31] measure the similarity based on the shortest paths in the graphs. Shervashidze et al. [125], on the other hand, count the number of graphlets, i.e. subgraphs with $k$ nodes. Note that these kernels can be defined both on unattributed and attributed graphs, where the attributes are generally on the nodes. Another interesting approach is that of Bai and Hancock [20], where the authors investigate the possibility of defining a graph kernel based on the Jensen-Shannon kernel. The Jensen-Shannon kernel is a non-extensive information theoretic kernel, which is defined in terms of the entropy of probability distributions over the structures being compared [96]. Bai and Hancock extend this idea to the graph domain by associating with each graph either its Von Neumann entropy [108], i.e., the Shannon entropy associated with the Laplacian eigenvalues of the graph, or the steady state distribution of a random walk on the graph. For a more complete review of graph kernels see [142].

A similar but somehow simpler problem that we need to face in the graph domain is that of measuring the similarity, or alternatively the distance, between graphs. Generally, the similarity between two graphs can be defined in terms of the lowest cost sequence of edit operations, for example, the deletion, insertion and substitution of nodes and edges, which are required to transform one graph into the other [121]. Another approach is that of Barrow and Burstall [24], where the similarity of two graphs is characterized using the cardinality of their maximum common subgraphs. Similarly, Bunke and Shearer [37] introduced a metric on unattributed graphs based on the maximum common subgraph, which later Hidović and Pelillo extended to the case of attributed graphs [70, 140]. Unfortunately, both computing the graph edit distance and finding the maximum common subgraphs turn out to be computationally hard problems. For an extensive review of graph distance measures refer to [61].

## 2.4 Quantum Computation

In quantum information, the quantum bit, or *qubit*, is the fundamental unit of information, and represents the quantum analogue of the classical bit. Just as a classical bit can be in a state, i.e., 0 or 1, a qubit can be in the state $|0\rangle$ or $|1\rangle$, where the Dirac notation is used as the standard notation for quantum mechanics. In contrast with a classical bit, however, a qubit can be also in a *superposition* of these two states, i.e., a linear combination of the form

$$\left|\psi\right\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.1}$$

where $\alpha$ and $\beta$ are complex *probability amplitudes* such that $|\alpha|^2 + |\beta|^2 = 1$. In other words, the state space for a qubit is a ray in a Hilbert space.

Quantum algorithms have gained a lot of popularity due to the possibility of exploiting quantum-mechanical phenomena such as superposition and *entanglement* in order to obtain consistent speedups over classical computers. However, In the next two Sections we will focus our attention on the relevant literature on quantum walks and divergence measure between quantum states, as we will make extensive use of these concepts in Chapter 5 and Chapter 6. For a comprehensive book on quantum computing and quantum information the reader is referred to [105].

### 2.4.1 Quantum Walks

Recently, there has been an increasing interest in using quantum walks as a primitive for designing novel quantum algorithms [79, 15, 42, 117] on graph structures. Quantum walks on graphs represent the quantum mechanical analogue of the classical random walk on a graph. Despite being similar in the definition, the dynamics of the two walks can be remarkably different. This is due mainly to the fact that while the state vector of the classical random walk is real-valued, in the quantum case the state vector is complex-valued. This property allows different paths of the walk to interfere with each other in both constructive and destructive ways. In the classical case the evolution of the walk is governed by a double stochastic matrix, while in the quantum case the evolution is governed by a unitary matrix, thus rendering the walk reversible. This in turn implies that the quantum walk is non-ergodic and, most importantly, it does not have a limiting distribution. Fig. 2.6 shows the distribution of a quantum walk and a random walk at time $T$ on a line. Quantum walks have been extensively studied on a wide variety of graphs [99, 81], such as the infinite line, cycles, regular lattices, star graphs and complete graphs. Because of these properties, quantum walks have been shown

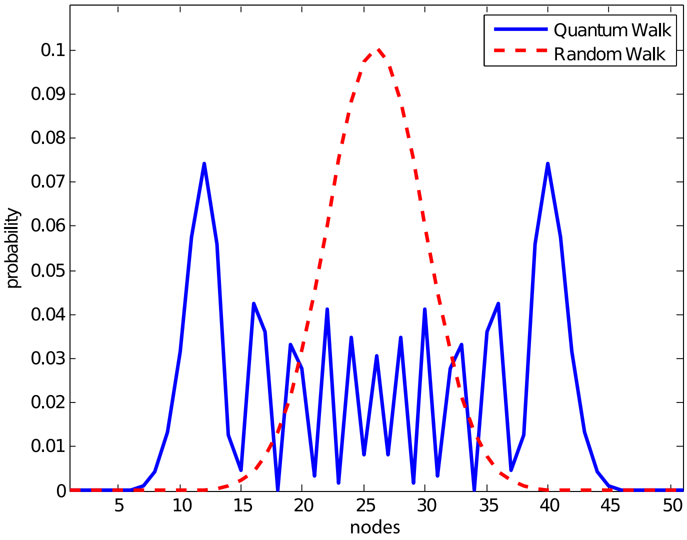


Figure 2.6: Probability distributions of a quantum walk (blue) and a random walk (red) on a line. Note that the quantum walk is spreading faster than the random walk.

to outperform their classical analog in a number of specific tasks, leading to polynomial and sometimes even exponential speedups over classical computation [122, 55]. For example, Farhi and Gutmann [55] have shown that if we take two co-joined $n$-level binary trees that are connected by their leaves, a quantum walk commencing from the root of the first tree can hit the root of the second tree exponentially faster than a similarly defined classical random walk. The major contribution of Farhi and Gutmann's work [55] is to show that one may achieve an exponential speedup without relying on the quantum Fourier transform.

In the case of the co-joined trees graph described above, the presence of a sym-

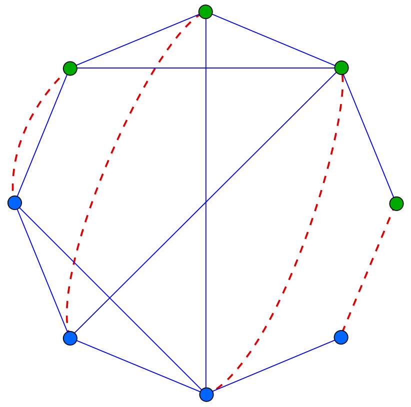


Figure 2.7: An example of a graph displaying a symmetrical structure, where we highlighted the pairs of symmetrical vertices. Note that by permuting the pairs of linked nodes the adjacency relations are preserved.

metrical structure is of key importance to the speedup. Given a graph $G = (V, E)$, an automorphism is a permutation $\sigma$ of the set of vertices $V$ of the graph which preserves the adjacency relations, i.e. if $(u, v) \in E$ then $(\sigma(u), \sigma(v)) \in E$. The set of symmetries of $G$ can thus be represented by its automorphism group $\mathrm{Aut}(G)$. Figure 2.7 shows an example of a symmetric graph. Whenever the graph possess some kind of symmetry, the constructive interference between certain paths will lead to faster hitting times. A number of recent works have further investigated the connection between the structural symmetries of the graph and the evolution of the quantum walk. For instance, Krovi and Brun [84] have proved that the phenomenon of infinite hitting times is generally a consequence of the symmetry of the graph and its automorphism group. Emms et al. [52] showed that there is a link between symmetries in the graph structure and a quasi-quantum analogue of the commute time. Specifically, the authors define a quasi-quantum analogue of the commute time associated with the continuous-time quantum walk and then explore the possibility of using it to embed the nodes of the graph into a low dimensional vector space. Their work reveals that the symmetries of the graph correspond to degenerate directions in the quantum commute time embedding space. However, their analysis is not based on a principled observable and is hence semi-classical.

### 2.4.2 Divergence Measures

In the context of quantum computation and quantum information, a number of distance (divergence) measures have been introduced in the literature. One of the reasons that makes these measures particularly attractive is the possibility of discriminating between different quantum states. In his seminal paper, Wootters [148] investigates the problem of distinguishability and defines the concept of statistical distance between pure quantum states. Wootters' work is fundamentally based on the extension of a distance over the space of probability distributions to the Hilbert space of pure quantum states. Similarly, attempts to define a distance measure between pure and mixed quantum states are typically based on the generalization of divergence or distance measures commonly used in the space of probability distributions, as it is the case for the Hellinger distance [69]. The same holds for the relative entropy [89], which is a generalization of information theoretic Kullback-Leibler divergence. However, note the relative entropy is neither a distance, as it is not symmetric, nor does it not satisfy the triangle inequality, and, most importantly, it is unbounded.

A more recent case is that of the quantum Jensen-Shannon divergence [93, 94, 86]. The classical Jensen-Shannon divergence [88] is a measure of similarity between probability distributions that has its routes in information theory. Unlike the Kullback-Liebler divergence [85], it is both symmetric and is directly linked to a metric (it is the square of a metric). Moreover, it can be used to define positive semi-definite kernels. As a result, the underlying metric space of probability distributions can be isometrically embedded in a real valued Hilbert-space. The quantum Jensen-Shannon divergence has recently been developed as a generalization of classical Jensen-Shannon di-

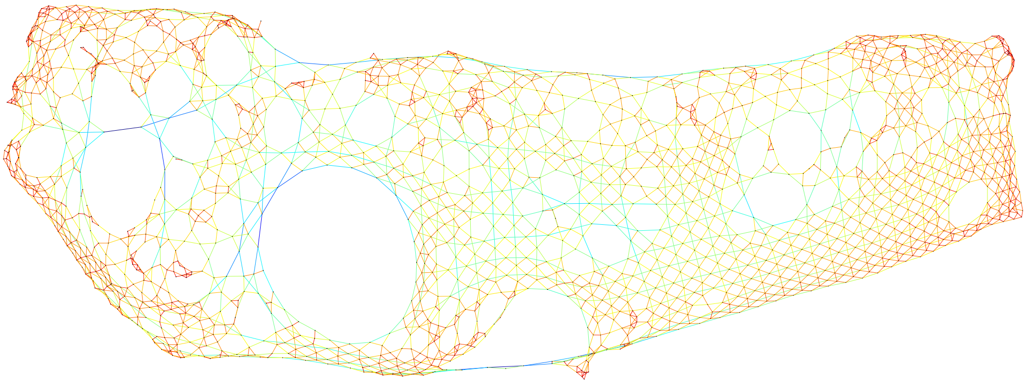


Figure 2.8: The road network of the city of Hollywood is an example of a complex system that can be naturally represented using a graph structure.

vergence to quantum states by Majtey, Lamberti and Prato [93, 94, 86]. The QJSD is computed from the density matrices defined using the outer-products of the eigenvectors of the quantum systems being compared. As a result the QJSD is given as the difference in Von Neumann entropy [105] of the mixed and pure states. For pure states the square root QJSD is proved to be a metric, while for mixed states there is strong experimental evidence that it is. Moreover, the authors show that for mixed quantum states the quantum Jensen-Shannon divergence has good distinguishability properties.

## 2.5 Graph Structure Analysis

Complex networks are usually defined as graphs with non-trivial topological features that describe the interactions between a set of entities. The study of complex networks [54] has recently attracted considerable interest because of the large variety of complex systems that can be modeled and analyzed using graphs. A non-exhaustive list of examples includes metabolic networks [73], protein interactions [72], brain networks [131], vascular systems [145], scientific collaboration networks [103] and road maps [77]. Fig. 2.8 shows a graph representation of the road network of the city of Hollywood. Properties such as small-worldness and the power-law distribution of vertex degrees [54] have been observed in several real-world networks, suggesting a marked difference with Erdös-Rényi random graphs [53].

An important area of research in complex network science deals with the spectral properties of graphs. An undirected graph $G(V,E)$ is usually represented in terms of its symmetric adjacency matrix

$$A_{uv} = \begin{cases} 1 \text{ if } (u,v) \in E \\ 0 \text{ otherwise} \end{cases} \tag{2.2}$$

where $V$ is the set of $n$ nodes of the graph and $E = V \times V$ is the set of edges. Let $D$

be the diagonal matrix with elements $d_u = \sum_{v=1}^{n} A(u,v)$, where $d_u$ is the degree of the node $u$. Then, we define $L = D - A$ as the graph Laplacian, a combinatorial analogue of the Laplace-Beltrami operator [75]. The spectrum of the adjacency or Laplacian matrix of a graph is then defined as the set of eigenvalues $\lambda_i$, each associated with an eigenvector $\phi_i$. Although for large graphs the complete eigendecomposition of the adjacency matrix may prove to be computationally too expensive, we can get some interesting insights into the structure of the graph by looking only at a small sample of the eigenvalues. For instance, one can show that the size of the maximum clique in a graph is at least $n/(n-\lambda_1(A))$, where $\lambda_1(A)$ denotes the larges eigenvalue of the adjacency matrix. More generally, spectral graph theory proves that to some extent it is possible to distinguish among different types of structures simply by looking at their spectrum. For an extensive study of the relation between the spectra of adjacency matrices and the structure of graphs refer to [27, 43].

More recently there has been some interest in characterizing the presence of symmetries in graphs [91] [149]. Recall that, given a graph $G(V,E)$, an automorphism is a permutation $\sigma$ of the set of vertices $V$ of the graph which preserve the adjacency relations, i.e. if $(u,v) \in E$ then $(\sigma(u),\sigma(v)) \in E$. Hence we can view the group of automorphisms $\mathrm{Aut}(G)$ of a graph as a representation of its symmetries. MacArthur et al. [91] observe that many real-world graphs possess a very large automorphism group, in contrast to classical random graph models. In particular the authors observe the presence of a certain number of small symmetric subgraphs, such as tree-like or clique-like structures, and relate this to the redundancy and thus robustness of real-world graphs. Note however that the problem of finding the set of automorphisms of a graph is actually an instance of the graph isomorphism problem, and thus it belongs to the NP class. Xiao et al. [149] study the origin of symmetry in real-world graphs. In common with [91], their work is based on the analysis of local symmetric motifs such as symmetric bicliques, i.e. an induced complete bipartite subgraph, denoted as $K_{V_1,V_2}$, in which every vertex of $V_1$ is connected to every vertex of $V_2$. Their analysis reveals that the symmetry of graphs is a consequence of a particular linkage pattern, where vertices with similar degrees tend to share common neighbors. It is also worth mentioning the work of Mowshowitz [98], which links the complexity of a graph to the entropy of the distribution of symmetric orbits.

Another fundamental task in graph structure analysis is that of measuring the importance of a vertex. To this end, a large number of centrality indices have been introduced in the literature [54], and each of these measures capture different but equally significant aspects of a vertex importance. The most common examples are probably the degree, closeness and betweenness centrality [57, 58, 104].

The degree centrality is defined as the number of links incident upon a node. Given a graph $G$ with $n$ nodes and adjacency matrix $A$, the degree centrality of $u$ is

$$DC(u) = \sum_{v=1}^{n} A(u,v) \tag{2.3}$$

The degree centrality naturally interprets the number of edges incident on a vertex as a measure of its “popularity". Alternatively, it can be interpreted as the risk of a node being infected in a disease spreading scenario.

The closeness centrality links the importance of a vertex to its proximity to the remaining vertices of the graph. More precisely, the closeness centrality is defined as the as the inverse of the sum of the distance of a vertex to all other nodes of the graph,

$$CC(u) = \frac{n-1}{s(u)} \tag{2.4}$$

where $s(u)$ denotes the sum of the distances from $u$ to all the other nodes of the graph, i.e.,

$$s(u) = \sum_{v=1}^{n} d(u,v) \tag{2.5}$$

where $d(u,v)$ denotes the distance between $u$ and $v$.

Finally, the betweenness centrality is a measure of the extent to which a vertex lies on the paths between others, where the path may be either the shortest path or a random walk between the nodes. If $sp(v_1, v_2)$ denotes the number of shortest paths from node $v_1$ to node $v_2$, and $sp(v_1, u, v_2)$ denotes the number of shortest paths from $v_1$ to $v_2$ that go through $u$, the betweenness centrality of $u$ is

$$BC(u) = \sum_{v_1=1}^{n} \sum_{v_2=1}^{n} \frac{sp(v_1, u, v_2)}{sp(v_1, v_2)} \tag{2.6}$$

Note that this definition assumes that the communication takes place along the shortest path between two vertices. A number of measures have been introduced to take into account alternative scenarios in which the information flows through different paths [54, 57, 58, 104].

# 3

# Medial Surfaces Extraction

The skeleton has proven to be a valuable and widely used shape descriptor for a number of tasks such as 2-D and 3-D shape recognition [110, 129], volumetric models deformation [28, 150], segmentation [114] and protein structure identification [13]. The interest in this descriptor stems from its being a concise representation of the original shape, which is topologically equivalent to it, and invariant to several shape deformations.

When working in two dimensions, the skeleton, or medial axis transform, is defined as the locus of the centers of the maximal inscribed circles bitangent to the shape boundary. Alternatively, it can be defined as the set of singularity points created by the inward evolution of the shape boundary with constant velocity according to the eikonal equation $\frac{\vec{C}(t)}{dt} = v\vec{N}(t)$, where $\vec{C}(t)$ is the equation of the boundary at time $t$, $v$ is the constant velocity and $\vec{N}(t)$ is the normal to the boundary. Finally the skeleton can be seen as the set of ridge points of the distance map [29] [41], where the distance map is the function $D(x, y)$ that assigns to every point in the interior of a shape its distance to the closest point on the boundary.

Our purpose in this Chapter is to propose a novel algorithm for medial surfaces extraction that is based on a generalization to three dimensions of the density-corrected analysis of Torsello and Hancock [138], while taking an adaptive octree-based approach for the discretization of the initial mesh in a manner that is similar to that proposed by Bai et al. [21] and Quadros et al. [111]. Contrary to these approaches, we decide not to precompute the whole octree in advance, but instead we keep the original mesh, that is used for distance computations, and we iteratively decide whether if refining a voxel or not based on the local value of the divergence of the momentum field, i.e., the confidence we have in that point being skeletal. Finally we design a simple alignment procedure to correct the displacement of the extracted skeleton with respect to the true underlying medial surface. We evaluate the proposed approach with an extensive series of qualitative and quantitative experiments, comparing our method against other approaches in the literature under varying mesh conditions.

## 3.1 Preliminaries

In this Section we review the two-dimensional continuous formulation of the Hamilton-Jacobi skeleton [127] and its density corrected counterpart [138], where the latter will form the basis for our medial surface extraction algorithm.

### 3.1.1 Hamilton-Jacobi Skeleton

Let the distance map $D$ be a function that assigns to each point in the interior of the shape its distance to the closest point on the object boundary $\vec{C}$. The velocity field is defined as

$$\vec{F} = \nabla D \tag{3.1}$$

where $\nabla = (\partial/\partial x, \partial/\partial y)^T$ is the gradient operator. Under the assumption that the vector field $\vec{F}$ is conservative everywhere except on the skeleton, the skeletal points can be identified by looking for those points where the system ceases to be conservative. Note that in this setting, the flux of the vector field $\vec{F}$ can be seen as modeling the flow of an incompressible fluid. Since the net flux of $\vec{F}$ through the boundary of the shape is positive, by virtue of the divergence theorem it follows the interior of the shapes contains a set of points which behave like sinks, i.e., the skeletal points. Recall that the divergence is defined as

$$\nabla \cdot \vec{F} = \lim_{|A| \to 0} \frac{\int_{\partial A} \vec{F} \cdot \vec{n}\, dl}{|A|} = \lim_{|A| \to 0} \frac{\phi_A(\vec{F})}{|A|} \tag{3.2}$$

where $A$ is an arbitrary area, $\partial A$ is its boundary, $\vec{n}$ is the outward norm at each point on $\partial A$ and $\phi_A(\vec{F})$ is the net flux of $\vec{F}$ through $\partial A$. That is, the divergence of $\vec{F}$ is proportional to the net flux of the field $\vec{F}$. Hence, in [127] the authors propose to identify as skeletal those points where

$$\lim_{|A| \to 0} \frac{\phi_A(\vec{F})}{|A|} < 0 \tag{3.3}$$

or, using (3.2),

$$\nabla \cdot \vec{F} < 0 \tag{3.4}$$

It can be shown, however, that the flux of $\vec{F}$ is not conservative, and as a consequence $\vec{F}$ can be seen as modeling the flow of a compressible rather than an incompressible fluid. When the fluid is compressible, however, its density changes during the inward evolution in a way which is proportional to the boundary curvature, and as a result the velocity field is no longer conservative.

As a first attempt to overcome this problem, one may introduce the normalized flux

$$N\phi_A(\vec{F}) = \lim_{r \to 0} \frac{\phi_A(\vec{F})}{2\pi r} \tag{3.5}$$

Since $\phi_A(\vec{F}) = \nabla \cdot \vec{F}(\xi)|A|$ where $\xi \in A$ and $|A| = \pi r^2$, in the limit the normalized flux becomes

$$\lim_{r\to 0} \frac{\phi_A(\vec{F})}{2\pi r} = \lim_{r\to 0} \frac{\nabla \cdot \vec{F}(\xi)}{2} r = 0 \tag{3.6}$$

Note, however, that due to the discrete structure of the image lattice the integration radius can only theoretically approach zero, and its minimum value will be of one pixel. More precisely, Torsello and Hancock [138] show that, assuming an integration radius of one pixel, the normalized flux at $p$ is

$$N\phi_A(\vec{F})(p) = -\frac{k(p)}{2} \tag{3.7}$$

where $k(p)$ is the curvature at a location $p$ belonging to the evolving boundary. As a result, near the endpoints of the skeleton, due to the extreme curvature of the boundary front, the value of the flux will tend to infinity, causing severe problems in the extraction of the skeleton.

### Density-Corrected analysis

Consider a now segment $dl(t)$ of the boundary front $\vec{C}(t)$ at time $t$. Assume that $dl(t)$ has average linear density $\rho(t)$ and length $l(t)$. Now, under the eikonal equation, at time $t+\Delta t$ the segment will evolve into $dl(t+\Delta t)$ with average linear density $\rho(t+\Delta t)$ and length $l(t+\Delta t)$. Assuming that the boundary front is curved, this implies $l(t) \neq l(t+\Delta t)$. Also, since the mass $m = l(t)\rho(t)$ of the segment is conserved, we have that $l(t)\rho(t) = l(t+\Delta t)\rho(t+\Delta t))$ and thus $\rho(t) \neq \rho(t+\Delta t)$.

In other words, when the front is curved the average linear density is not constant over time and thus we have to resort to the more general principle of conservation of mass. Based on this intuition, Torsello and Hancock [138] suggest that there is indeed a conservative field associated with the dynamics of the boundary evolution, but this cannot be the velocity field. Instead, they define the so-called *momentum field* $\vec{M} = \rho\vec{F}$, where $\rho$ is a scalar field that assigns to each point the linear density of the boundary front. By a simple analysis of the change of density of $dl(t)$ over the time, one can prove that

$$\nabla \cdot (\rho\vec{F}) = 0 \tag{3.8}$$

as suggested by the above intuition, and thus $\phi_A(\rho\vec{F}) = 0$ for any region $A$ not containing a skeletal point.

Finally, applying the rule of product differentiation to the conservation equation and setting $\sigma = \log(\rho)$ we obtain

$$\nabla\sigma \cdot \vec{F} = -\nabla \cdot \vec{F}, \tag{3.9}$$

which can be further reduced to the system of ordinary differential equations along the path of boundary points

$$\begin{cases} \frac{\partial}{\partial t}\sigma(s(t)) = -\nabla \cdot \vec{F}(s(t)) \\ \frac{\partial}{\partial t}s(t) = \vec{F}(s(t)) \end{cases} \tag{3.10}$$

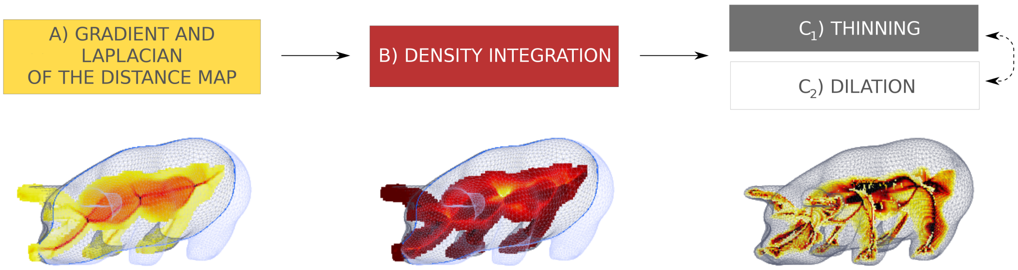


Figure 3.1: Steps to refine the skeleton: a) computation of the gradient and Laplacian of the distance map; b) integration of the log-density in the voxels with a full neighborhood; c) alternating thinning and dilation step to detect skeletal voxels at the current level of the octree.

where $s(t)$ is the trajectory of a boundary point under the eikonal equation.

## 3.2 Hierarchical Skeletonization

Our algorithm works as follows. We are given a triangulated mesh, a starting resolution $res_{min}$ and a desired resolution $res_{max}$. Initially we compute a complete voxelization of the shape at resolution $res_{min}$. Given this initial coarse discretization, we compute the distance transform $D$, its gradient $\vec{F} = \nabla D$ and the divergence $\nabla \cdot \vec{F}$, then we integrate the density $\sigma = \log(\rho)$ and finally we compute the divergence of the momentum field $\nabla \cdot (\rho \vec{F})$. With this information to hand, we are able to extract a first approximation of the medial surface. Assuming that a very low starting resolution $res_{min}$ is given as input, we now wish to further refine the extracted skeleton up to a $res_{max}$ resolution.

To this end, we iteratively increase the resolution by subdividing the leaves of the octree with a large value of $\nabla \cdot (\rho \vec{F})$, i.e., those voxels that are most likely to contain skeletal points. The Hamiltonian analysis is then carried over the newly created octree level and the refinement process is iterated until the required resolution $res_{max}$ and octree level $\log_8(res_{max})$ is reached.

In order to carry over the Hamiltonian analysis at a lower octree level the following steps must be undertaken (see Fig. 3.1):

1. **Velocity field computation**. For each voxel $\vec{v}$ at the current resolution level we compute its distance to the shape boundary. Given the distance map, we first compute its gradient in $\vec{v}$ by fitting a hyperplane in a least squares sense on the voxel neighbors, then we determine its Laplacian by computing the flux of $\vec{F}$ through the surface of the convex-hull bounded by the neighbours of $\vec{v}$, divided by its volume.

2. **Integration of the front-density**. For each voxel at the current resolution level we compute the density of the evolving front by evaluating Eq. 3.10. We integrate the

density starting from the current level boundary inward, under the assumption that the initial boundary has a complete 26-neighborhood where the value of the density is inherited from the parent voxels.

3. **Thinning and dilation**. With the divergence information to hand, we iteratively remove the current level boundary voxels in distance order when the value of the divergence is under a certain threshold. In order to guarantee the preservation of the object topology, we remove a voxel only if it is simple, i.e., if its removal does not alter the object topology by disconnecting the shape or introducing a hole. Once the thinning procedure is completed, we dilate the skeleton to partially compensate for discretization errors incurred at the coarser levels. We alternate the thinning-dilation process until no voxels can be added to the thinned skeleton. Finally a last dilation is performed to guarantee that the exploded points have a complete neighborhood around each skeletal point.

With this high-level overview in mind, we will now present all the computational ingredients needed by the proposed approach.

### 3.2.1 Distance Computation

The distance transform computation is certainly one of the most expensive operations that we need to perform. We decide not to compute the distance map with respect to a discretized boundary, instead we keep the original mesh and we make distance queries with respect to it. In particular, the input mesh is saved on an Axis Aligned Bounding Box (AABB) tree, a common data structure that is used to make distance queries faster. A voxel is assigned either to the interior or exterior of the shape by casting a ray from the center of the voxel to a random direction and computing the number of intersections with the mesh. If the number of intersections is odd, the point is classified as interior, otherwise it is classified as exterior. We acknowledge that better algorithms for computing the signed distance transform have been proposed in the literature (e.g., [19]), but we also want to stress that the distance map issue is completely incidental to the main problem of skeletonization, which is the one we are addressing in this Chapter.

### 3.2.2 Gradient and Laplacian Computation

Once the distance map is to hand, its gradient and divergence can be determined. Note, however, that while in the beginning all the leaves of the octree are at the same level and thus the gradient and the Laplacian can be approximated using the finite difference method, as the skeleton is refined there will be several voxels at different levels of resolution. For this reason we need resort to a different approximation method that is able to cope with a non-uniform grid setting.

Note that in the remainder of the Chapter we will operate on different neighborhoods of a voxel, according to the type of operation that we intend to perform. This include the 6−, 18− and 26− neighborhoods, where $n-$ refers to the adjacency relation between the voxels. Recall that two voxels are 6-adjacency if they share a face, 18-adjacent if they share a face or an edge and 26-adjacent if they share a face, an edge or a vertex. In particular, we will always assume that a 26-neighborhood is used, with the exception of a few cases. As explained later in the text, when computing the laplacian of the distance map we only use local information and thus we restrict ourselves to a 6-neighborhood. On the other hand, during the integration of the density, we will use the subset of the 26-neighbors that have already been visited by the inward-evolving boundary. Finally, when ensuring the topology preservation, we will refer to the work of Malandain et al. [95], where the 6−, 18− and 26− neighborhoods are used to characterize the voxels.

Following [102], we compute the gradient by performing a 4D linear regression over all the neighbors of $\vec{x}$. More formally, given a set of points $\{(x_i, y_i, z_i, d_i)\}_{i=1}^{m}$, where $(x_i, y_i, z_i)^T$ is a neighbor of $\vec{x}$ and $d_i$ its distance to the boundary, we look for the coefficients $A, B, C, D$ minimizing

$$E(A,B,C,D) = \sum_i w_i (Ax_i + By_i + Cz_i + D - d_i)^2. \tag{3.11}$$

The gradient is then $\vec{F}(\vec{x}) = \frac{(A,B,C)^T}{||(A,B,C)^T||}$. As a weight $w_i$ we used the inverse of the distance of the point $(x_i, y_i, z_i)^T$.

Note that this approach has a problem whenever the skeleton crosses the convex hull of the neighborhood, as we integrate across a singularity resulting in erroneous computation of the gradient. A common solution to this problems to perform one-sided computations to avoid crossing the singularity, however one-sided computations usually exhibit larger bias. Here we chose to perform a two-sided computation of the gradient as we are not interested in its value close to the singularity as we are adopting a one-sided process for the computation of the momentum field. The experiments will show, that even with this possible instability due to the possibility of crossing a singularity in the computation of the gradient, the momentum filed is well conserved outside the of the skeletal branches resulting in a well localized skeleton.

As for the laplacian of the distance map, i.e., the divergence of the velocity field, we compute it using a discretization of the divergence theorem around the convex hull of the 6-neighborhood of each point. Note that even if the leaves are not guaranteed to be at the same level, and thus we cannot guarantee to have a complete 26- or 18- neighborhood, due to the octree construct we always have at least a 6-neighborhood. Doing a linear approximation of $\vec{F}(\vec{x})$ over the faces of the convex hull, we can approximate the flux

$$\Phi_U(\vec{x}) = \int_{\delta U} \vec{F}(s) \cdot \vec{n}(s)\, ds \approx \sum_{t=1}^{8} \frac{1}{3} A_t \vec{n}_t \cdot \left( \sum_{\vec{p} \in V_t} \vec{F}(\vec{p}) \right), \tag{3.12}$$

where $U$ is the convex hull of the 6-neighbors of $\vec{x}$ and $A_t$, $\vec{n}_t$, and $V_t$ are respectively the area, the normal, and the set of vertices of the (triangular) faces of $U$. Due to the divergence theorem, we have $\int_U \nabla \cdot \vec{F}(\vec{x})\, dx = \Phi_U(\vec{x})$, from which we obtain the following

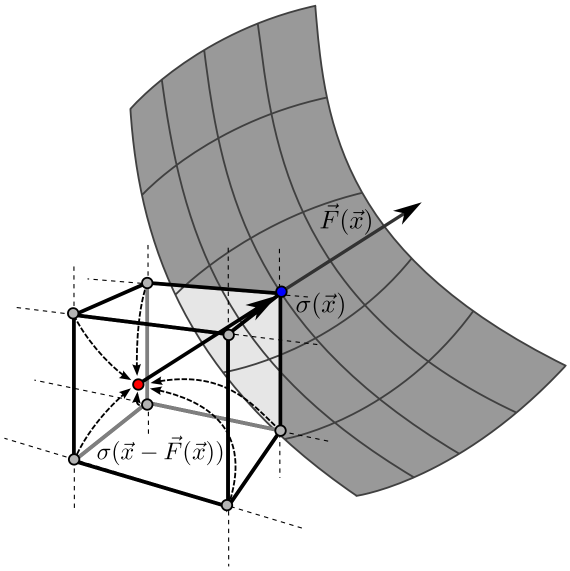


Figure 3.2: Integration of the density along the boundary path.

discretization for the divergence:

$$\nabla \cdot \vec{F}(\vec{x}) \approx \frac{\Phi_U(\vec{x})}{|U|} \approx \frac{\sum_{t=1}^{8} \frac{1}{3} A_t \vec{n}_t \cdot \left(\sum_{\vec{p} \in V_t} \vec{F}(\vec{p})\right)}{|U|}. \tag{3.13}$$

### 3.2.3 Integration of the Momentum Field

Once the distance, gradient and Laplacian have been computed, we can integrate the density in the newly subdivided skeletal points.

It is of key importance that the density integration is carried out only on those point that have a complete 26-neighborhood, i.e., those with a homogeneous neighborhood. The voxels with a non-homogeneous neighborhood, on the other hand, will simply inherit the value of the density and divergence fields of their parent node. The reason for this is that an inhomogeneous neighborhood induces a higher discretization error to the direction of the gradient which will severely affect the accuracy of the integration step. Thus, before refining the skeleton to a higher resolution level, we perform a dilation of the skeletal voxels in order to guarantee that all their children will indeed have a complete neighborhood. Then, after the refinement, there will be a 1-voxel thick boundary of voxels with non-homogeneous neighborhood that will be children of the dilation voxels, rather than of the skeletal voxels. Note that this dilation can simply be considered a part of the last thinning/dilation step of the refinement of the previous level, which will be described later.

In order to compute the momentum field over the interior of the shape we need to solve Eq. 3.10. A common approach in this case is that of solving the linear system obtained by rewriting Eq. 3.10 as a system of difference equation. The problem here is that the skeleton is a set of singularities of momentum field, i.e., we expect the density field to have different values at opposite sides of a medial surface. Consequently, the linear system has no solution. Even looking for an approximate solution using a gradient descent method would result in oscillations near the skeleton, so a different approach is needed.

As proposed by Torsello and Hancock [138], we decide to integrate the equation in the time domain. The critical point is to ensure that when we compute the log-density $\sigma$ of boundary points at time $t$ we reference only the values of $\sigma$ calculated at points already crossed by the inward-evolving boundary. In order to do so, we opt to find a numerical solution of Eq. 3.10 using a Crank-Nicolson approximation [47].

Assume that there exists a family of surfaces $\vec{B}_t$ representing the inward evolution of the boundary $\vec{B}$, that can be locally parametrized as $\vec{B}_t(u,v)$ around any point $\vec{x}$. Then, we have

$$\sigma(\vec{B}_t(u,v)) = \sigma(\vec{B}_{t-1}(u,v)) + \frac{1}{2}[\nabla\cdot\vec{F}(\vec{B}_t(u,v)) + \nabla\cdot\vec{F}(\vec{B}_{t-1}(u,v))] \tag{3.14}$$

In the spatial domain, if $\vec{x} = \vec{B}_t(u,v)$ we have $\vec{B}_{t-1}(u,v) \approx \vec{x} - \vec{F}(\vec{x})$, which, substituted into Eq. 3.14, yields

$$\sigma(\vec{x}) = \sigma(\vec{x} - \vec{F}(\vec{x})) + \frac{1}{2}[\nabla\cdot\vec{F}(\vec{x}) + \nabla\cdot\vec{F}(\vec{x} - \vec{F}(\vec{x}))] \tag{3.15}$$

Unfortunately the point $\vec{x} - \vec{F}(\vec{x})$ is not guaranteed to belong to the cubic lattice, so we actually need to interpolate it using the values at the eight vertices of the cube containing it. Once again we should ensure that the interpolation doesn't cross the medial surfaces. Luckily, $\vec{x}$ is the last of the eight vertices visited by the evolving boundary, so this requirement is met. Thus we can safely use the trilinear interpolation which yields

$$\begin{aligned}\sigma(\vec{x}) = \Big(&\sigma(\vec{x} - \vec{F}(\vec{x})) - (1-|F_1|)(1-|F_2|)(1-|F_3|)\sigma(\vec{x})\\ &+\tfrac{1}{2}[\nabla\cdot\vec{F}(\vec{x}) + \nabla\cdot\vec{F}(\vec{x} - \vec{F}(\vec{x}))]\Big)/(1-(1-|F_1|)(1-|F_2|)(1-|F_3|))\end{aligned} \tag{3.16}$$

where, $F_1$, $F_2$, and $F_3$, are the three components of $\vec{F}(\vec{x})$ and, due to the fact that we use trilinear interpolation, $\sigma(\vec{x}-\vec{F}(\vec{x})) - (1-|F_1|)(1-|F_2|)(1-|F_3|)\sigma(\vec{x})$ does not depend on the value of $\sigma(\vec{x})$. As Fig. 3.2 shows, the point $\vec{x} - \vec{F}(\vec{x})$ does not to the belong to the cubic lattice. We then interpolate it using the values of the log-density on the eight corners of the cube containing the point. Note that $\vec{x}$ is the last of the eight vertices which is visited during the boundary evolution, and thus we are guaranteed that all the points that we use for the interpolation are on the same side of the medial surface.

Given this formulation, we can integrate the value of the log-density over the interior of the shape, starting from the most external voxels inwards. At the first level the most external voxels will be the boundary boxes, which have a unit density, and thus a null log-density. At all other steps, the external voxels will be the voxels with irregular

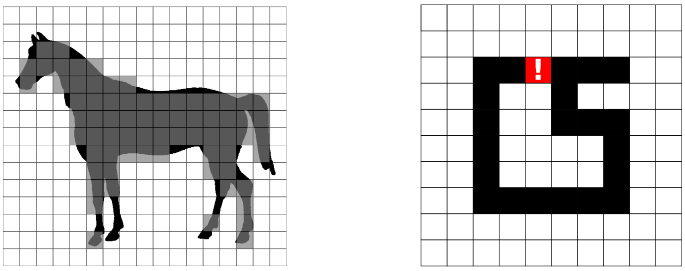


Figure 3.3: The dilation process is needed to regain detail lost at lower levels, although care must be given not to change the shape topology.

neighborhood that inherit the log-density from their parents. Once the log-density has been integrated, we can proceed to compute the divergence of the momentum field in each point of the interior of the shape. The value of $\nabla \cdot (\rho\vec{F})(\vec{x})$ is given by approximating Eq. 3.9 as follows

$$\begin{aligned}\nabla \cdot (\rho\vec{F})(\vec{x}) &= \Delta\sigma e^{\sigma(\vec{x})-\frac{1}{2}\Delta\sigma} \\ &+\frac{1}{2}\left[\nabla \cdot \vec{F}(\vec{x}-\vec{F}(\vec{x}))e^{\sigma(\vec{x}-\vec{F}(\vec{x}))} + \nabla \cdot \vec{F}(\vec{x})e^{\sigma(\vec{x})}\right]\end{aligned} \tag{3.17}$$

where $\Delta\sigma = \sigma(\vec{x}) - \sigma(\vec{x}-\vec{F}(\vec{x}))$. Note that, since the equations introduced in this Section are to be evaluated at different levels of resolution, the integration step is actually dependent on the corresponding voxel size.

### 3.2.4 Skeleton Extraction

With the divergence information to hand, we can select the voxels that are likely to contain skeletal points and that will be further subdivided to form the next level in the octree. The skeleton extraction is based on a thinning process guided by the value of the divergence of the momentum field at each voxel.

**Divergence Driven Thinning**

In [137] Torsello and Hancock show that the field $\rho\vec{F}$ is conservative outside skeletal branches, while its flux through a 1-voxel circle centered on a skeletal point is proportional to $\mathrm{d}l/\mathrm{d}s$, i.e., the ratio between the boundary length $\mathrm{d}l$ and the skeletal segment length $\mathrm{d}s$. This means that theoretically, skeletal branches can be detected by checking voxels with negative divergence of the momentum field. However, adopting any spatial discretization to compute the flux results in a spread-out of the divergence-based signal.

Following Torsello and Hancock, we thin the shape by iteratively removing boundary points in decreasing order of divergence. That is to say that without any further control on the thinning process we might actually end up introducing holes in the skeleton or even splitting it into disjoint parts.

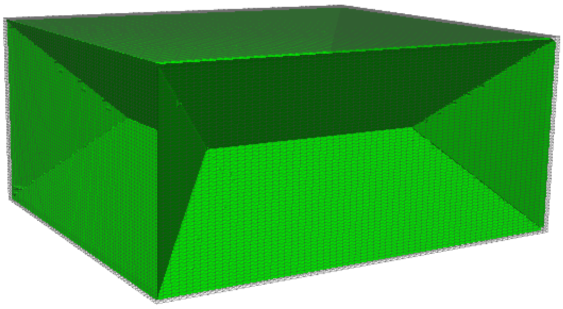


Figure 3.4: A box shape and its medial surface.

Recall that one of the key properties of the skeleton is that of having the same topology of the original shape. While for some approaches like the Voronoi-based ones this comes at no cost, the voxel-based methods should always take into account whether if the removal of a voxel would disconnect the shape, introduce a hole or erode it by deleting the endpoints. Unfortunately, when dealing with volumetric objects, ensuring that this property holds is not always an easy task. Hence, in this Chapter we resort to the voxel classification of Malandain et al. [95], which allows us to efficiently identify removable voxels by exploring the connectivity of their neighborhood. More precisely, Malandain et al. show how to classify a 3D point $\vec{x}$ in a cubic lattice by computing two features. Let $N_n(\vec{x})$ denote the $n$-adjacent neighbors of $\vec{x}$. Then $C^*(\vec{x})$ and $\bar{C}(\vec{x})$, defined as follows.

$C^*(\vec{x})$ is the number of the 26-connected components 26-adjacent to $\vec{x}$ in $B \cap N^*_{26}(\vec{x})$, where $B$ is the set of object points.

$\bar{C}(\vec{x})$ is the number of the 6-connected components 6-adjacent to $\vec{x}$ in $W \cap N_{18}(\vec{x})$, where $W$ is the set of background points.

With this result to hand, we can easily identify the simple points of the medial surface [95], i.e., those points whose removal does not alter the topology of the object. We can then proceed with the thinning process by iteratively removing all simple points in decreasing order of divergence. More precisely, the conditions for a point to be removed are that 1) it is simple, 2) it is not an endpoint and 3) it is characterized by a negative divergence of the momentum field. Note, however, that due to the errors introduced by the discretization of the shape, after the first thinning process the medial surface can be two-voxel thick in certain regions. To ensure thinness at the highest resolution level we further thin the shape by removing all those points that are simple but not endpoints of the surface, regardless of their divergence. Following [129], we decide to restrict our definition of an endpoint to a 6-neighborhood. In this case, it can be shown that a necessary condition for a point to be an endpoint is to have three 6-adjacent background voxels

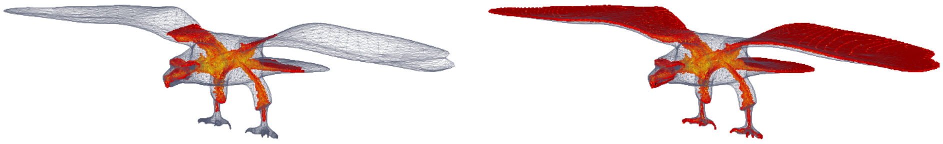


Figure 3.5: Dilating the skeleton recovers details lost in the coarser levels.

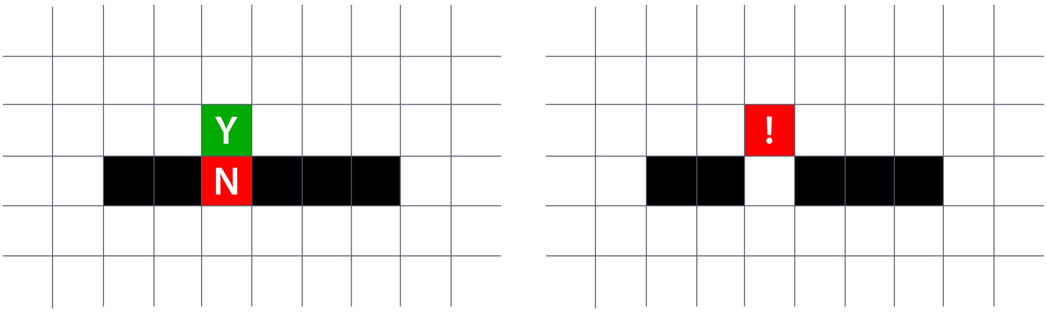
### Skeleton Dilation

With the proposed hierarchical approach, once a voxel is flagged as non skeletal at any level, all its descendants will inherit the property. A problem with this is that fine details might be lost at coarser level, resulting in parts of the skeleton that will be missing at all levels (see Fig. 3.3). Further, note that the skeletal voxels detected at the coarsest level are not even guaranteed to be connected and, since all further processing is topology preserving, a disconnected skeleton will remain disconnected at all levels.

We address the latter problem by keeping only the largest component, while the missing detail is addressed by dilating the skeleton after it has been computed at each new level. This way, once the voxels are small enough to capture the detail, the skeleton will regrow into the missing parts. Note that since the dilation adds new voxels to the current medial surface, we need to ensure that the topology is preserved, thus we dilate only into voxels that would become simple after the dilation (see Fig. 3.3).

Let $V$ denote the set of voxels before we start thinning the current level of the tree, and let $U$ be the subset of $V$ formed by the boundary voxels of $V$. We then thin $V$ to reveal the skeletal voxels as previously described. After the thinning step, we check if some voxel $v \in U$ has been selected as skeletal. If that is the case, we dilate it and we compute $D$, $\vec{F}$, $\nabla \cdot \vec{F}$, $\rho$, $\nabla \cdot (\rho\vec{F})$ on the dilated set. Then, we apply the thinning process again. The dilation-thinning process is iterated until the thinned skeleton contains no boundary voxels. This process gives us an adaptive dilation which adds only new candidate skeletal voxels with a large value of $\nabla \cdot (\rho\vec{F})$ and thus can be skeletal. Fig. 3.4 shows the special case of a box shape, together with the extracted medial surface. Initially, the whole set of voxels in the interior of the cube belongs to $V$, while the boundary voxels on the faces, edges and vertices of the cube belong also to $U$. Because of the negative value of the divergence, the voxels on the edges of the cube will survive the first thinning step, and thus will be selected as skeletal. Since these voxels belong to $U$, they will be dilated, as explained above. Note that $U$ will also be updated in order to include the new dilated boundary of $V$. However, the following thinning iteration will remove all the voxels in $U$, and the dilation-thinning process will finally converge. Note that during all these steps we always ensure that the topology of the object is not altered by adding or removing only simple points.

With this improvement, we are able to recover small details that might have been lost during the first discretizations, as well as longer skeletal segments. Fig. 3.5 shows how critical this procedure is. The eagle model in the figure clearly needs a very dense voxelization in order to capture details such as the claws, or even entire parts such



(a) Before Thinning (b) After Thinning

Figure 3.6: The final iteration of the thinning procedure removes all the simple points which are not endpoints. In this way, however, it can introduce small bumps on the surface, as shown in (b). Here we would like to remove the vertex marked with **Y**, but since this voxel satisfies the endpoint condition it cannot be deleted.

as the wings. With the proposed approach, one can simply start from a lower and less computationally intensive resolution and then refine the extracted skeleton to a certain desired resolution.

Finally, once the iterated dilation-thinning process gives us the final skeleton, we perform one final dilation step to ensure the presence of a complete 26-neighborhood around the new set of voxels on which we need to compute $\rho$ and $\nabla \cdot (\rho\vec{F})$. At the last resolution level, the final dilation process is substituted with the endpoint-driven thinning that gives us a 1-voxel thick medial surface.

### 3.2.5 Medial Surface Alignment

At the end ot the thinning process, we obtain the set of voxels most likely to contain the medial axis, thus placing vertices at the center of the voxels, and deriving the mesh connectivity from the adjacency information of the voxels, will result in a fine approximation of the medial surface in the form of a triangulated mesh. There are, however two sources of noise that limit the quality of the extracted surface, but that can effectively be addressed with a post-processing step.

The first is an artifact due to the limited control over the order in which the thinning process eliminates the voxels. The final iteration of the thinning procedure removes all the simple points which are not endpoints, however, thinning order, and the topology and endpoints preservation rules might prevent us from choosing the correct skeletal voxels as candidate for elimination, while preferring some adjacent voxel which are not endpoints and whose removal doesn't alter the object topology (see Fig. 3.6). As a consequence, depending on the spatial order of the thinning, we might introduce little bumps on the surface. Due to their formation process, these bumps can be detected easily by comparing their distance to the surface to that of a nearby voxels. Let $d(v)$

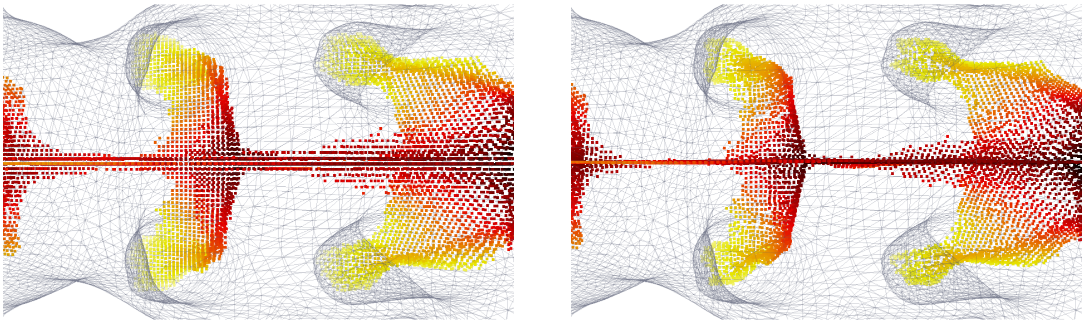


Figure 3.7: Due to the voxelization, the centers of the voxels are very likely to be displaced with respect to the true underlying medial surface. Hence the medial surface alignment procedure is needed to achieve a better approximation of the skeleton.

be the distance of candidate point $v$ from the shape's surface, let $\vec{F}(v)$ be the gradient of the distance map in $v$, and let $w$ be the neighbor of $v$ in the direction of $\vec{F}(v)$, i.e., closest to the line $v + t\vec{F}(v)$. If $d(w) > d(v)$ then we $v$ is a bump and we simply remove $v$ from the set of skeletal voxels and mark $w$ as skeletal.

The second limit is a result of the discrete nature of the grid: the centers of the skeletal voxels will be actually slightly displaced with respect to the true underlying medial surface. We address this issue by allowing the final vertices to move within the voxel from the central position to one that is most likely to lie in the skeletal surface, resulting in a higher precision skeletal mesh even at low voxel resolution (see Fig. 3.7).

Hence, given a voxel $v$, we compare the orientation of its velocity field (gradient of the distance transform) with that of its 26-neighbours, in order to determine which voxels lie on the other side of the medial surface. We call this set $O_v$. Note that thanks to the previous refinement step, we are sure that at least one of $v$'s neighbours will indeed lie on the other side of the medial surface. With the set of voxels to hand, we proceed by computing for each voxel $w \in O_v$ belonging to this set the intersection between the true medial surface and the line connecting $w$ and $v$. Let $s_v$ and $s_w$ be the surface points closest to $v$ and $w$ respectively, we look for the point $p_w = \alpha v + (1-\alpha) w$ along the line connecting $v$ to $w$, for which $||p_w - s_v|| = ||p_w - s_w||$, i.e., is equidistant from the closest surface points. This point $p_w$ is likely to be very close to the medial surface, but it displacement from the original position is not limited to the direction of inward motion of the surface and has also a tangential component. We eliminate this by interpolating the position over all the neighbors in $O_v$.

Fig. 3.8 illustrates the interpolation process. Let $O_v = \{w_1, \cdots, w_k\}$ and let $p_1, \cdots, p_k$ be the corresponding estimated points on the medial surface, we interpolate between their position using Shepard's inverse distance weighting method [123]. Shepard's interpolation method is a generalized barycentric interpolation approach designed for sparse data. It reconstruct the position of a point as a linear combination of the sam-

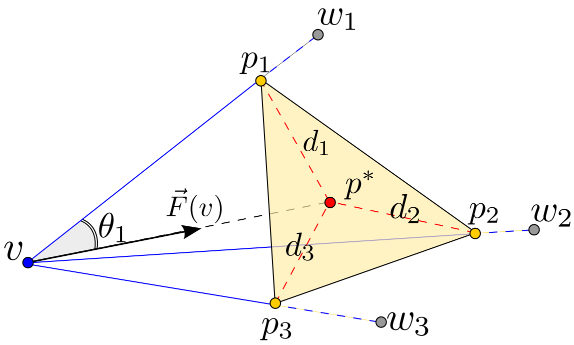
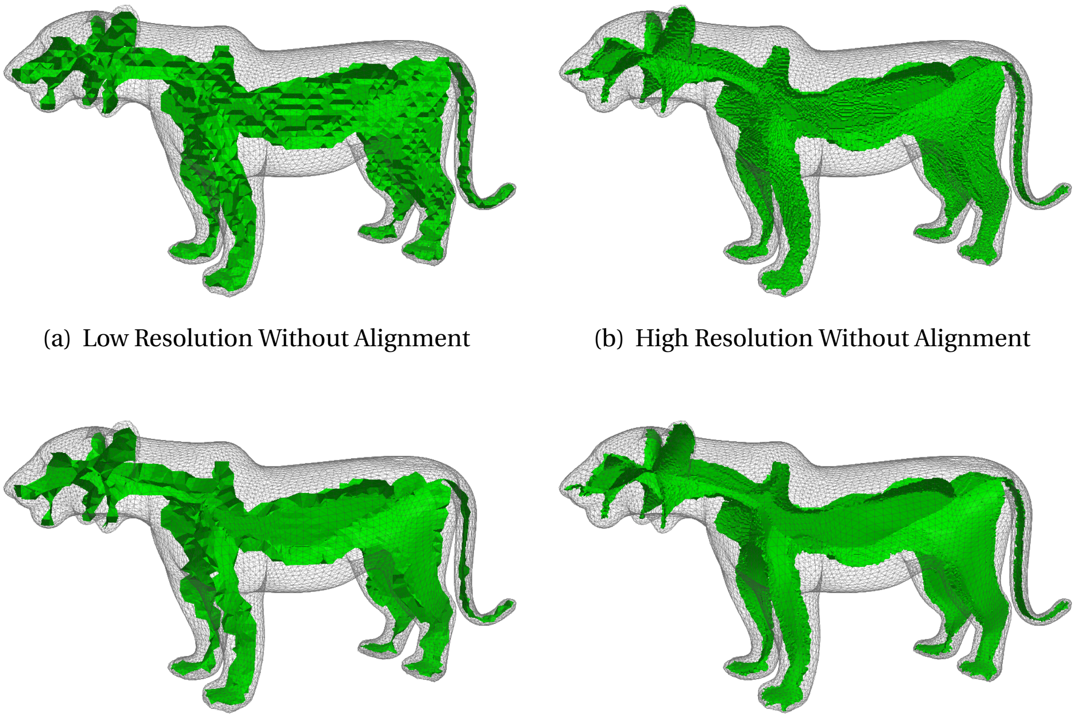


Figure 3.8: The location of the realigned skeletal point is estimated performing an inverse-distance weighted interpolation of the points $p_i$ obtained finding the bitangent point along the lines connecting $v$ to its neighbors on the other side of the skeletal surface.



(a) Low Resolution Without Alignment

(b) High Resolution Without Alignment

(c) Low Resolution With Alignment

(d) High Resolution With Alignment

Figure 3.9: The proposed alignment procedure yields a faster convergence speed, in the sense that we are able to get a good approximation of the real underlying medial surface even at low levels of resolution.

ples $p_i$

$$p^* = \frac{\sum_{i=1}^{k} w_i p_i}{\sum_{i=1}^{k} w_i} \tag{3.18}$$

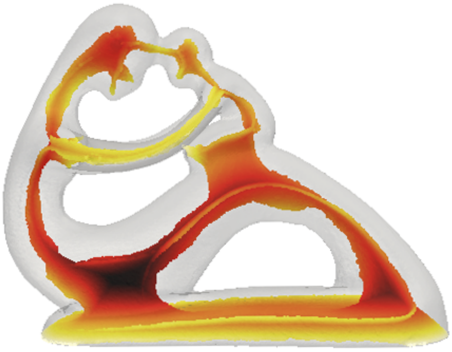


Figure 3.10: The medial surface of a shape with genus greater than 0.

where the weights $w_i$ are a function of the inverse distance $d_i$ of the interpolant $p^*$ to the samples $p_i$, usually $w_i = \frac{1}{d_i^2}$.

In order to apply Shepard formula we need to estimate the (squared) distances of the points $p_i$ to the interpolant $p^*$. To this end we make the simplifying assumption that the gradient of the distance map $\vec{F}$ is approximately orthogonal to the medial surface at $p^*$. Under this assumption we note that $d_i = ||p_i - v|| \sin \theta_i$, where $\theta_i$ is the angle between $\vec{F}(v)$ and $v\vec{p}_i$, and thus

$$w_i = \frac{1}{d_i^2} = \frac{1}{||p_i - v||^2 \sin^2 \theta_i} = \frac{1}{||p_i - v||^2 (1 - \cos^2 \theta_i)} = \\ \frac{1}{||p_i - v||^2 - \left((p_i - v)^T \vec{F}(v)\right)^2}. \quad (3.19)$$

Fig. 3.7 shows the result of the alignment procedure on the voxels of a medial surface segment. Perhaps the major advantage of the proposed procedure is that it yields a faster convergence speed for the medial surface extraction algorithm. Fig. 3.9 clearly shows that when we skip the alignment step we need to increase the depth of the hierarchical refinement considerably in order to get a decent approximation of the underlying medial surface. On the other hand, if we align the skeletal voxels as described in this Section we can stop the hierarchical refinement earlier and still get a good result.

## 3.3 Experimental Results

In this Section we evaluate the quality of the proposed algorithm with a wide series of experiments. Here we present quantitative and qualitative comparison with three different approaches, namely the Hamilton-Jacobi algorithm of Siddiqi et al. [127], the multiscale algorithm of Reniers et al. [115] and the Voronoi-based approach of Yoshizawa et al. [150]. Note that the first two methods work on a voxelized 3D shape, while the latter works directly on the mesh. The analysis has been performed on a selection of 40 shapes from the Princeton Shape Benchmark [126] and the SHREC 2010 database [33]. All skeletons are extracted with $res_{min} = 16$ and $res_{max} = 1024$, unless otherwise stated. Note that the proposed approach works independently of the shape's genus, and our dataset include shapes with genus greater than zero (see for example Fig. 3.10).

Fig. 3.11 shows the hierarchical discretization of a sample shape. As expected, the deepest leaves, i.e., the highest resolution voxels, are located around the skeleton of the shape. Note also that the density of the voxelization is increased where the shape is more detailed, which is exactly what we expect to happen. This figure clearly shows the advantages of a hierarchical discretization of the shape against a complete one. The skeletons extracted at various stages of hierarchical refinement are shown in Fig. 3.12.

### 3.3.1 Qualitative Evaluation

Here we propose a qualitative evaluation of our algorithm by comparing it with the Voronoi-Based approach of Yoshizawa et al. [150], the Multiscale algorithm of Reniers et al. [115] and the standard Hamilton-Jacobi method. Both the implementations of [150] and [115] were downloaded from the authors websites, while we implemented the Hamilton-Jacobi algorithm simply by dropping the density integration procedure in our framework.

Fig. 3.13 shows a qualitative comparison between the four methods. The Voronoi skeleton is clearly the noisiest one and in most cases fails to provide an acceptable approximation of the medial surface, although it is computationally significantly less expensive than the other algorithms. The Multiscale approach on the other hand performs quite well, although due to the complexity of processing a complete voxelization of the shape it was not able to reach the level of detail of our method. Finally, the Hamilton-Jacobi skeletons exhibit a few spurious skeletal segments due to the lack of the correction of the curvature effects. Fig. 3.14 provides a magnified view of the torso and head of a selected medial surface extracted with our algorithm and the standard Hamilton-Jacobi method, respectively. As Fig. 3.14(b) shows, the head of the human shapes contains some spurious segments which are located as expected in the areas

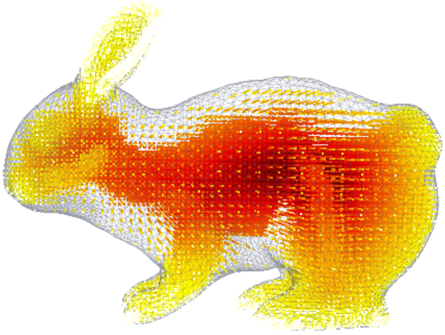
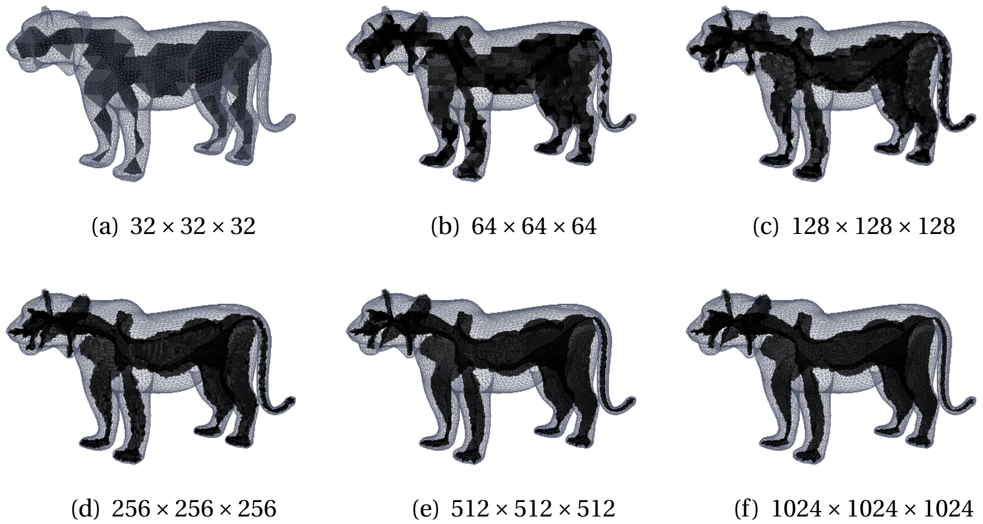


Figure 3.11: The shape voxelization performed by our algorithm: the highest resolution voxels are all located around the skeleton.


(a) 32 × 32 × 32 (b) 64 × 64 × 64 (c) 128 × 128 × 128

(d) 256 × 256 × 256 (e) 512 × 512 × 512 (f) 1024 × 1024 × 1024

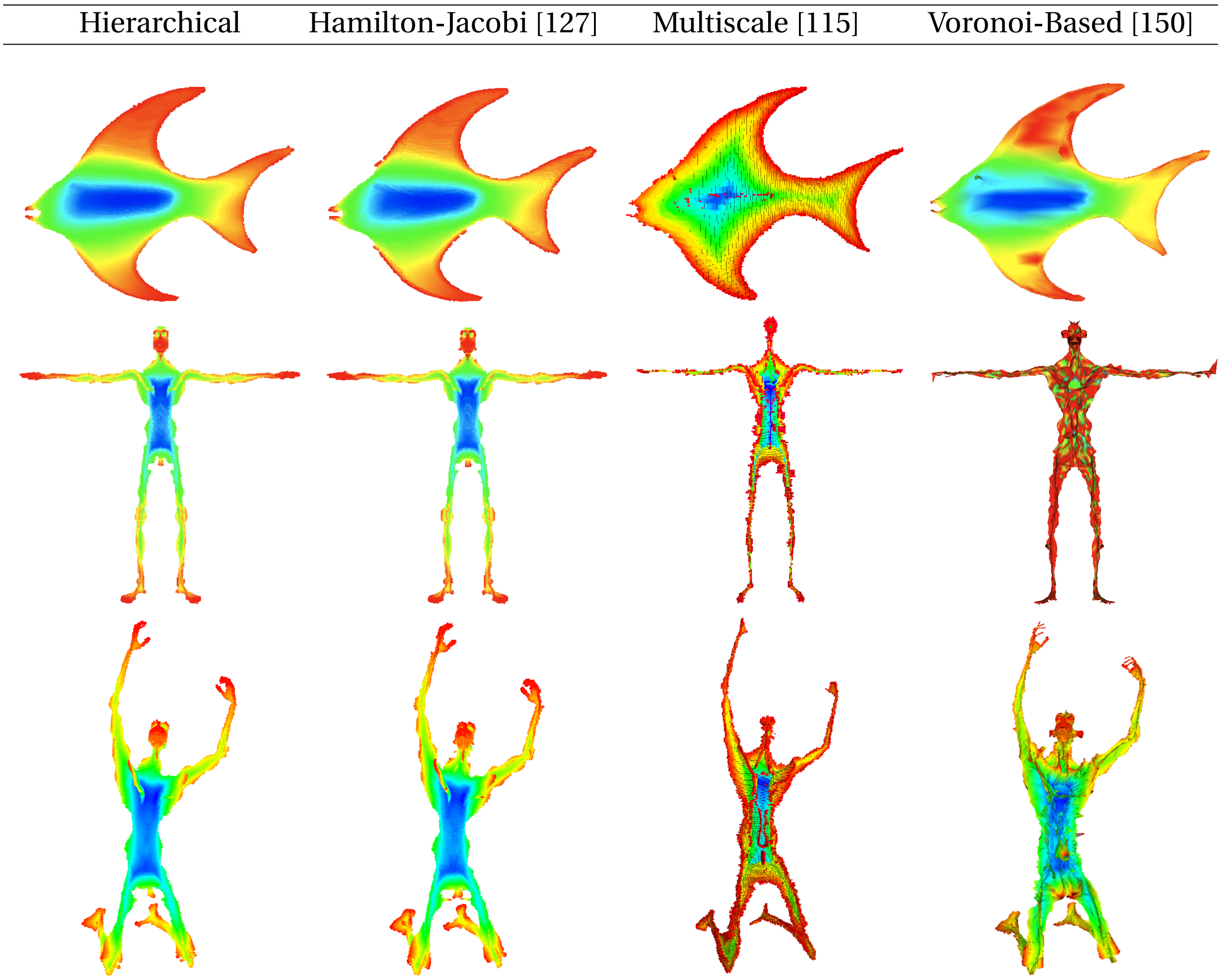
Figure 3.12: The hierarchical refinement of the medial surfaces. The skeletal points are meshed for ease of visualization.

of higher curvature. Although setting a stricter threshold eliminates these spurious branches, it also results in a loss of details in the torso, as highlighted in Fig. 3.14(c).

### 3.3.2 Skeleton Localization

The Hamilton Jacobi framework [127, 128] is based on the principle that the (normalized) flux around an infinitesimal area not containing a skeletal branch is zero, while it is non-zero over the skeleton. This guarantees the divergence-based thinning approach to converge to the exact location of the skeleton points. However, as noted in [138], this analysis is true only for the normalized flux and only in the limit. Adopting any spatial discretization to compute the normalized flux results in non-zero values also outside the skeleton that is proportional to the curvature of the inward evolving front. this results in a spread-out of the divergence-based signal especially close to skeletal endpoints, severely affecting the localization of the skeletal branches and also resulting in the creation of small spurious branches [138]. The curvature correction process [138], on the other hand, localized the non-zero values of the divergence much better, resulting in better localization and avoiding the creation of spurious branches.

In this Section we evaluate the localization properties of the skeletons extracted with our algorithm and we compare it against the standard Hamilton-Jacobi approach. To evaluate the localization properties of the density correction we plot the distribution of the voxels as a function of both divergence and distance to the skeleton. In order to evaluate the loss in localization caused by the hierarchical approach, we compare this distribution for shapes at the same target level but at different starting levels. In particular, the histograms in Fig. 3.15 plot the average distribution of skeletons ex-



Hierarchical Hamilton-Jacobi [127] Multiscale [115] Voronoi-Based [150]

Figure 3.13: Comparison of our approach against a standard Hamilton-Jacobi algorithm, the Multiscale algorithm of Reniers et al. [115] and the Voronoi-Based approach of Yoshizawa et al. [150].

tracted at the maximum resolution of 128 × 128 × 128, with starting resolutions going from 128 × 128 × 128 (single level), to 16 × 16 × 16 (multi-level (16)), thus all the skeletons were extracted with varying levels of hierarchical refinement.

First we note that when the hierarchical approach goes through more levels, the points tend to be more concentrated around the skeleton. This is to be expected since there is a decrease in the total number of voxels expanded. In general we see that the proposed algorithm yields a good localization of the skeleton, since the points with non-zero divergence are all located near the skeleton, while the points that are far from the skeleton have a value of the divergence equal to zero. However, we do observe a little noise due to the propagation of numerical errors, which is typical of hierarchical algorithms. Nonetheless, the distribution remains tightly peaked, with very few points far from the skeleton with a non-negligible divergence of the momentum field.

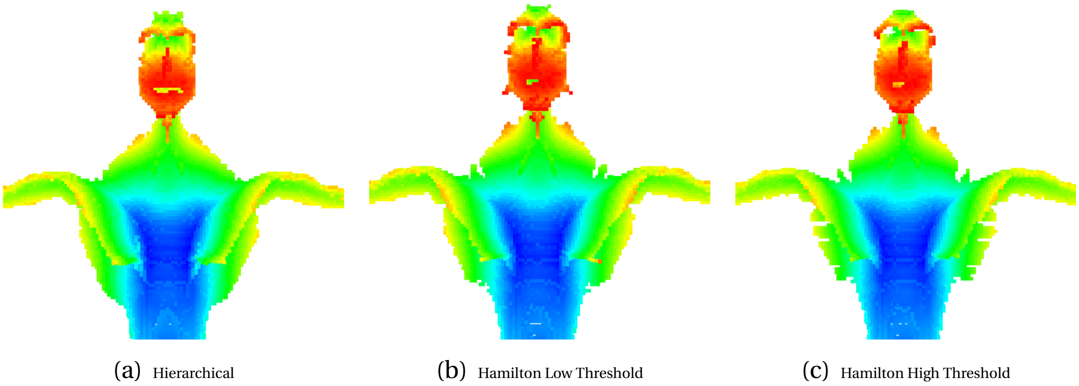

(a) Hierarchical (b) Hamilton Low Threshold (c) Hamilton High Threshold

Figure 3.14: A magnified view of the head and torso of the medial surface of a human shape. The standard Hamilton-Jacobi algorithm produces spurious segments which can be removed by setting a stricter threshold, although this results in a loss of details of the torso.

Fig. 3.16 compares the localization of the divergence of the momentum field against that of the velocity field as used by Siddiqi et al. [127]. As previously reported by Torsello and Hancock [138], even in 3D the momentum field localizes the skeleton much more tightly than the velocity field.

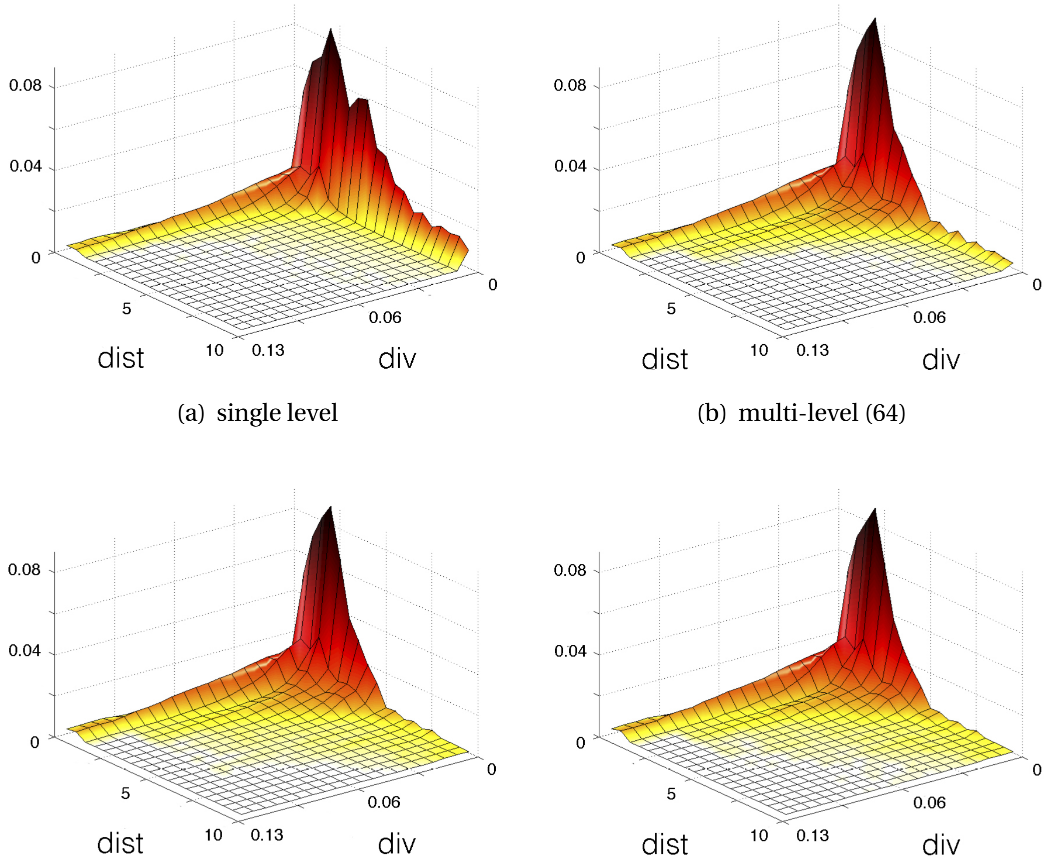
Here we show also a slice of the shape voxelization in order to reveal its interior, where the voxels are colored according to the value of the divergence, i.e., low values correspond to white while high (negative) values correspond to black. Recall that the value of $\nabla \cdot \vec{F}$ in a point $p$ depends on the local boundary curvature and thus its value tends to infinity as $p$ moves closer to a skeleton endpoint, even if $p$ is not skeletal.

As a consequence of this, we observe some blurred areas around the endpoints of the medial surface. On the other hand, in the density-corrected slice we see a much sharper localization of the skeleton.

### 3.3.3 Sensitivity to Mesh Resolution

We now evaluate the sensitivity of the proposed approach to different samplings and sampling densities of the mesh. Given a mesh, we compute 3 increasing simplifications where the number of triangles is decreased respectively to 50%, 25% and 10% (see Fig. 3.17). For each of these, we extract the medial surfaces using our approach, the standard Hamilton-Jacobi one, the Voronoi-Based approach of Yoshizawa et al. [150] and the Multiscale [115] algorithm. We then compute the average nearest neighbour distance between the voxels of the medial surfaces of the simplified meshes and those of the original medial surface.

Table 3.1 shows the average cost for different levels of simplification and different skeleton extraction methods. As we can see, our approach yields the minimum average distance, hence showing that it is less sensitive to the mesh resolution than the other



(a) single level

(b) multi-level (64)

(c) multi-level (32)

(d) multi-level (16)

Figure 3.15: Distribution of the voxels as a function of both divergence and distance to the skeleton. The starting resolution ranges from 128×128×128 to 16×16×16, while the maximum resolution remains fixed at 128 × 128 × 128. Note that the points with non-zero divergence are all located near the skeleton, while the points that are far from the skeleton have a value of the divergence equal to zero. We note a decrease of the total number of points that are located far from the skeleton, which is in line with the decrease of total voxels created. We also observe a little noise due to the propagation of numerical errors, which is typical of hierarchical algorithms.

methods. Note that under a 50% mesh simplification the Hamilton-Jacobi algorithm performs similarly to our method, as by removing 50% of the triangles the mesh quality is only slightly altered, and hence we don't observe the formation of new spurious branches. On the other hand, as we further simplify the mesh, its surfaces becomes less smooth and this in turns yields the formation of some spurious segments which induce a higher average nearest-neighbour distance. As expected, the Voronoi-based approach turns out to be the most unstable. It is known, in fact, that in the case of Voronoi-Based skeletonization algorithms the quality of the extracted medial surface

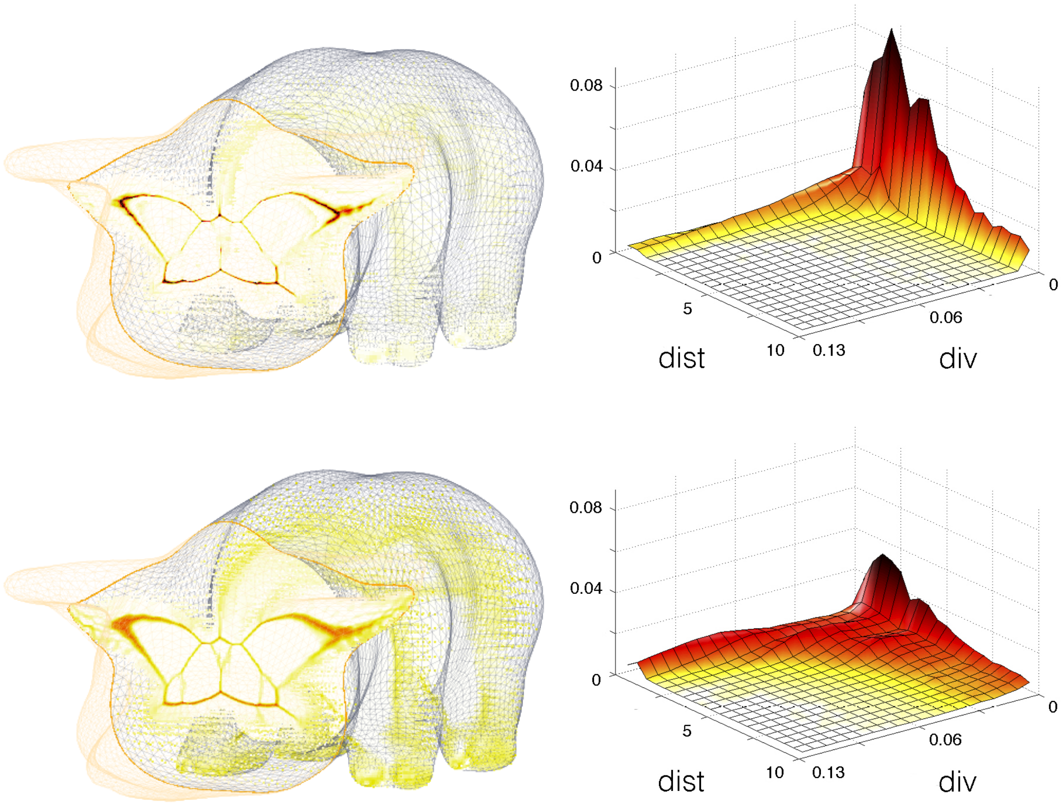


Figure 3.16: Comparison between the momentum field (left) and the velocity field (right). The left histogram shows a good localization of the skeleton, while in the right histogram we observe a non-negligible tail of distant points with non-zero divergence.

greatly depends on the mesh resolution and on how densely it is being sampled. It is hence clear that by simplifying the shape we are inevitably altering the quality of the resulting medial surface, as Table 3.1 clearly shows. Finally the Multiscale algorithm seems to perform slightly better than us when the number of triangles is decreased by 50%, while for higher levels of mesh simplification our approach is achieving better results.

### 3.3.4 Robustness Against Noise

A good skeletonization algorithm should also be able to deal with moderately noisy inputs. To this end, we approximate the skeletonization of the diffused shape by smoothing the distance map as in [138]. Hence, given a voxel and its neighborhood, we update the local value of the distance by interpolating the values of the distance function on its neighbors [76].

Fig. 3.18 shows the robustness to noise of the proposed approach. The results ob-

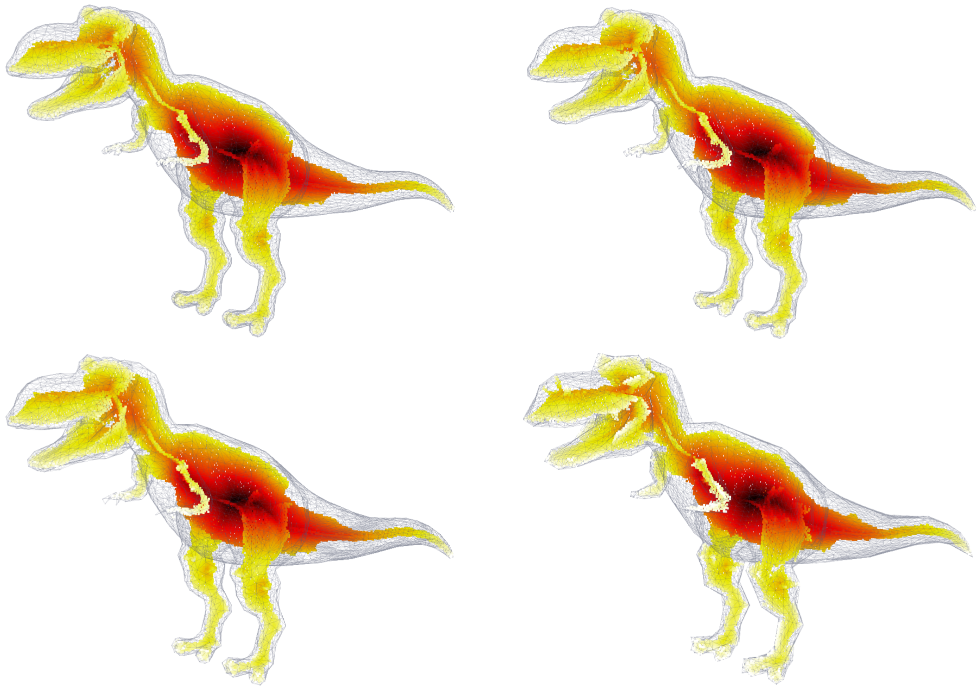


Figure 3.17: Medial surfaces of increasingly simplified meshes extracted, where the number of triangles is reduced to 50%, 25% and 10% respectively. All the medial surfaces are extracted using the proposed algorithm.

| **Mesh Simplification** | 50% | 75% | 90% |
|---|---|---|---|
| *Our Method* | 0.0009 | 0.0012 | 0.0017 |
| *Hamilton-Jacobi* | 0.0008 | 0.0014 | 0.0024 |
| *Multiscale* [115] | 0.0004 | 0.0019 | 0.0021 |
| *Voronoi-Based* [150] | 0.0032 | 0.0044 | 0.0051 |

Table 3.1: Average nearest neighbour distance between medial surface of the original shape and its simplified counterparts. Note that our methods is less sensitive to mesh quality when compared to the standard Hamilton-Jacobi approach, the Voronoi-Based approach of Yoshizawa et al. [150] and the Multiscale [115] algorithm.

tained by our algorithm and the Multiscale one are comparable. Note, though, that in the latter the robustness is achieved thanks to a fine tuning of the importance threshold, comes at the cost of losing some detail in the finer parts. On the other hand the Voronoi-based algorithm is unable to cope with the noise on the mesh boundary and thus performs much worse than the other approaches. Finally, the presence of noise clearly increases the formation of spurious branches in the Hamilton-Jacobi algorithm.

In order to evaluate quantitatively the robustness to noise, we compute again the average nearest neighbour distance between the medial surface extracted from the

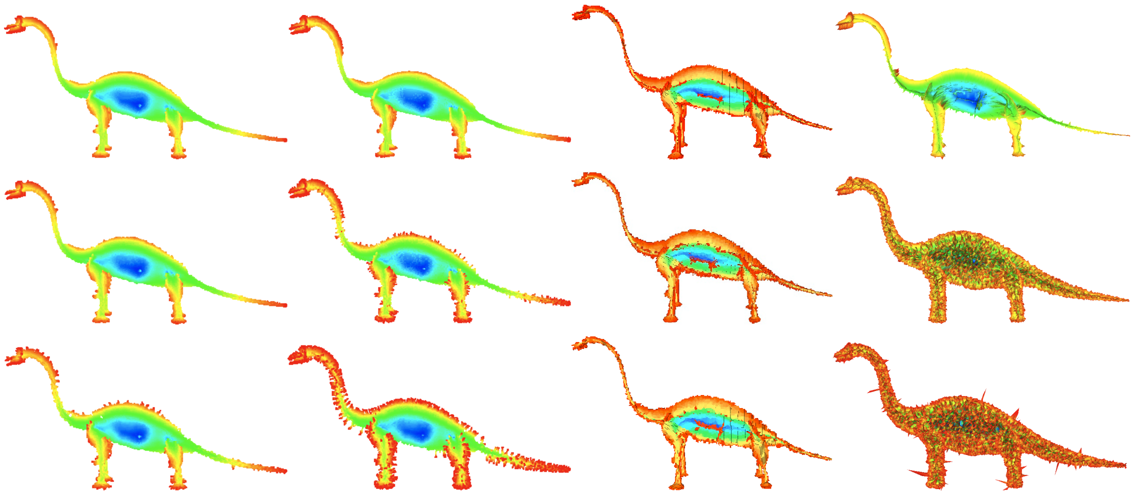
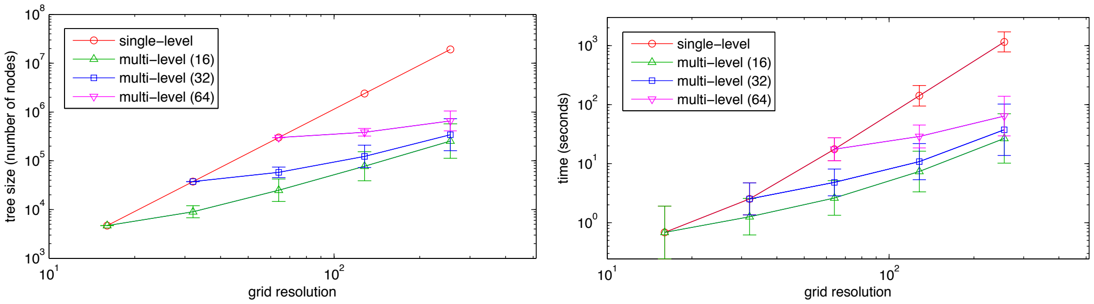


Figure 3.18: Effects of noise. The first row shows the skeletons extracted from the original object, while the second and the third rows show the skeletons after random vertex displacement of respectively 10% and 20% of the average edge applied to the shape. From left to right: our approach, Hamilton-Jacobi, Multiscale [115] and Voronoi-based [150].

original mesh and the medial surfaces extracted from the noisy shapes. The results are shown in Table 3.2. As the qualitative experiments suggested, the Voronoi-based approach is clearly performing worse than all the other methods, while the Multiscale approach and the proposed algorithm yield similar results, although we know that in the Multiscale approach this comes at the cost of losing fine details. Finally, once again the importance of the density correction is highlighted by the decreased performance of the standard Hamilton-Jacobi approach.

| **Mesh Noise** | 10% | 20% |
|---|---|---|
| *Our Method* | 0.0010 | 0.0014 |
| *Hamilton-Jacobi* | 0.0013 | 0.0033 |
| *Multiscale* [115] | 0.0009 | 0.0018 |
| *Voronoi-Based* [150] | 0.0112 | 0.0146 |

Table 3.2: Average nearest neighbour distance under increasing mesh noise. Compared to the standard Hamilton-Jacobi approach and the Voronoi-Based approach of Yoshizawa et al. [150], our methods is less sensitive to noise, while it performs similarly to the Multiscale [115] algorithm.



(a) memory requirement

(b) execution time

Figure 3.19: The plots show the memory and time requirements for the computation of a series of skeleton with different levels of refinement. Our approach clearly outperforms the standard algorithm where the space is completely discretized.

### 3.3.5 Time and Spatial Complexity

Perhaps the most obvious advantage of our algorithm is the decrease of space and time requirements. As for theoretical complexity, it is governed by the sorting of points with respect to their distance to the boundary that takes place before the density integration, which is $O(n \log(n))$, where $n$ is the number of leaves of the octree. Anyway, while in the case of a complete grid $n = m^3$, where $m$ is the final skeleton resolution, in the proposed approach the growth is only quadratic, i.e., $n = m^2$, since the voxels are refined only around the two-dimensional medial surfaces.

Fig. 3.19 shows the memory and time requirements for the extraction of a series of skeletons from a wide variety of shapes. Note that because of the higher memory requirements of the complete discretization, the machine on which the experiments were performed, which is equipped with 20 GB of RAM, couldn't afford resolutions beyond 256 × 256 × 256. On the other hand, using the hierarchical approach we could easily reach resolutions as high as 1024 × 1024 × 1024, which would have required 1,073,741,824 voxels if we were to voxelize the shape uniformly.

## 3.4 Conclusion

In this Chapter we presented a novel algorithm for medial surfaces extraction that is based on the density-corrected Hamiltonian analysis [138]. In order to cope with the exponential growth of the number of voxels, we compute a first coarse discretization of the mesh which is iteratively refined until a desired resolution is achieved. The refinement criterion relies on the analysis of the momentum filed, where only the voxels with a suitable value of the divergence are exploded to a lower level of the hierarchy. In order to partially compensate for the discretization errors incurred at the coarser levels, a dilation procedure is added at the end of each iteration. Finally we designed a simple alignment procedure to correct the displacement of the extracted skeleton with respect to the true underlying medial surface. We evaluated the proposed approach with an extensive series of qualitative and quantitative experiments.

With the skeleton to hand, one can segment it into different components and use a graph to represent the relation between these parts. However, this procedure is likely to introduce noisy nodes and edges in the graph. In the next Chapter we will show how to learn a generative model from a set of observed graphs which is able to capture the structural variations while designating an external node to model the presence of noise.

# 4 Learning Graph Structure

In the previous Chapter we have introduced a novel algorithm for the extraction of medial surfaces from 3D shapes. With the medial surface to hand, one can easily segment it into different components and use a graph to represent the relation between these parts. Given a set of graphs that represent our shape database, one is now faced with the problem of classifying these shapes into different objects.

Standard classification techniques can be usually divided into two broad categories, namely the generative and the discriminative approaches. Generally speaking, while generative approaches try estimate the joint probability density function $p(x, y)$ and obtain $p(y|x)$ by applying Bayes rule, discriminative approaches try to estimate $p(y|x)$ directly from the data, where $x$ denotes the data and $y$ denotes the class label. In this Chapter we propose a novel generative model for graphs which works by decoupling the structural and stochastic parts and making the naïve assumption that the observation of each node and each edge is independent of the others, but allows correlations to pop up by mixing different models. The model is described in Section 4.1, where we adopt a Minimum Message Length [143] criterion to prune mixture components and model nodes. An alternative criterion is proposed in Section 4.2, where we illustrate an adaptation of the Approximation Set Coding framework [35] to our model selection problem.

## 4.1 A Generative Model for Graphs

Consider the set of undirected graphs $S = (g_1, \dots, g_l)$, our goal is to learn a generative graph model $\mathcal{G}$ that can be used to describe the distribution of structural data and characterize the structural variations present in the set. To develop this probabilistic model, we make an important simplifying assumption: We assume that the model is a mixture of naïve models where observation of each node and each edge is independent of the others, thus imposing a conditional independence assumption similar to naïve Bayes classifier, but allowing correlation to pop up by mixing the models.

The naïve graph model $\mathcal{G}$ is composed by a structural part, i.e., a graph $G(V, E)$, and a stochastic part. The structural part encodes the structure, here $V$ are all the nodes that can be generated directly by the graph, and $E \subseteq V \times V$ is the set of possible edges. The stochastic part, on the other hand, encodes the variability in the observed

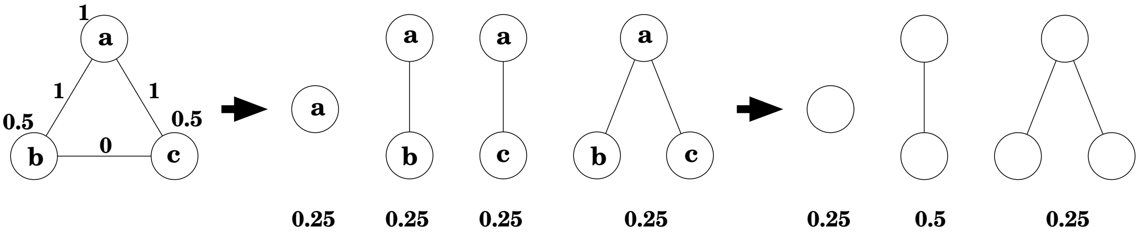


Figure 4.1: A structural model and the generated graphs. When the correspondence information is lost, the second and third graph become indistinguishable.

graph. To this end we have a series of binary random variables $\theta_i$ associated with each node and $\tau_{ij}$ associated with each edge, which give us respectively the probability that the corresponding node is generated by the model, and the probability that the corresponding edge is generated, conditioned on the generation of both endpoints. Further, to handle node- and edge-attributes, we assume the existence of generative models $W_i^n$ and $W_{i,j}^e$ that model the observable node and edge attribute respectively, and that are parametrized by the (possibly vectorial) quantities $\omega_i^n$ and $\omega_{i,j}^e$. Note that $\theta_i$ and $W_i^n$ need not be independent, nor do $\tau_{ij}$ and $W_{i,j}^e$. With this formalism, the generation of a graph from a naïve model is as follows: First we sample from the node binary indicator variables $\theta_i$ determining which nodes are observed, then we sample the variables $\tau_{i,j}$ indicating which edges between the observed nodes are generated, and finally we sample the attributes $W_i^n$ and $W_{i,j}^e$ for all observed nodes and edges, thus obtaining the full attributed graph.

Clearly, this approach can generate only graphs with fewer or equal nodes than $V$. This constraint limits the generalization capability of the model and forces one to model explicitly even the observed random isotropic noise. To correct this we add the ability to generate nodes and edges not explicitly modeled by the core model. This is obtained by enhancing the stochastic model with an external node observation model that samples a number of random *external nodes*, i.e., nodes not explicitly modeled in the generative model. The number of external nodes generated is assumed to follow a geometric distribution of parameter $1-\bar{\theta}$, while the probability of observing edges that have external nodes as one of the endpoints is assumed to be the result of a Bernoulli trial with a common observation probability $\bar{\tau}$. Further, we assume common attribute models $\bar{W}^n$ and $\bar{W}^e$ for external nodes and edges, parametrized by the quantities $\bar{\omega}^n$ and $\bar{\omega}^e$. This way external nodes allow us to model random isotropic noise in a compact way.

After the graph has been sampled from the generative model, we lose track of the correspondences between the sample's nodes and the nodes of the model that generated them. We can model this by saying that an unknown random permutation is applied to the nodes of the sample. For this reason, the observation probability of a sample graph depends on the unknown correspondences between sample and model

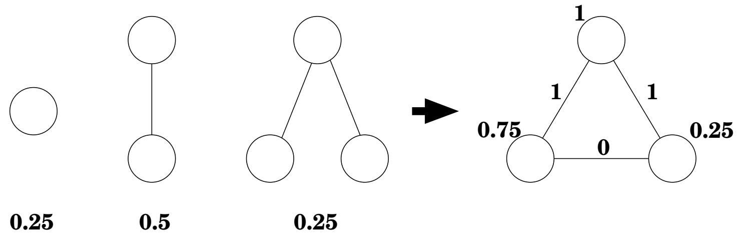


Figure 4.2: Model estimation bias. If a single node correspondence is taken into account the estimated model will exhibit a bias towards one of multiple possible correspondences.

nodes.

Figure 4.1 shows a graph model and the graphs that can be generated from it with the corresponding probabilities. Here model is unattributed with null probability of generating external nodes. The numbers next to the nodes and edges of the model represent the values of $\theta_i$ and $\tau_{i,j}$ respectively. Note that, when the correspondence information (letters in the Figure) is dropped, we cannot distinguish between the second and third graph anymore, yielding the final distribution. Note that, from a structural perspective, our generative model is essentially a complete graph, where each edge is labeled with a (possibly zero) probability of observation, in contrast to alternative approaches such as the generalized set median of Jiang et al. [74] and Ferrer et al. [56]. Also, while the latter aim at building a structural prototype of a set of observed graphs, our goal is that of learning a probabilistic model of the underlying structure.

Given the node independence assumptions at the basis of the naïve graph model, if we knew the correspondences $\sigma_g$ mapping the nodes of graph $g$ to the nodes of the model $\mathcal{G}$, we could very easily compute the probability of observing graph $g$ from model $\mathcal{G}$:

$$P(g|\mathcal{G},\sigma_g) = (1-\bar{\theta}) \prod_{i\in V} P(g_{\sigma_g^{-1}(i)}|\theta_i,\omega_i^n) \cdot \prod_{(i,j)\in E} P(g_{\sigma_g^{-1}(i),\sigma_g^{-1}(j)}|\tau_{i,j},\omega_{i,j}^e) \cdot$$
$$\cdot \prod_{i\notin V} P(g_{\sigma_g^{-1}(i)}|\bar{\theta},\bar{\omega^n}) \cdot \prod_{(i,j)\notin E} P(g_{\sigma_g^{-1}(i),\sigma_g^{-1}(j)}|\bar{\tau},\bar{\omega^e}),$$

where the indexes $i \in V$ and $(i,j) \in E$ indicate product over the internal nodes and edges, while, with an abuse of the formalism, we write $i \notin V$ and $(i,j) \notin E$ to refer to external nodes and edges. With the ability to compute the probability of generating any graph from the model, we can compute the complete data likelihood and do maximum likelihood estimation of the model $\mathcal{G}$, however, here we are interested in the situation where the correspondences are not known and must be inferred from the data as well.

Almost invariably, the approaches in the literature have used some graph matching technique to estimate the correspondences and use them in learning the model parameters. This is equivalent to defining the sampling probability for node $g$ as $P(g|\mathcal{G}) =$

$\max_{\sigma\in\Sigma_n} P(g|\mathcal{G},\sigma)$. However, as shown in [135], assuming the maximum likelihood estimation, or simply a single estimation, for the correspondences yields a bias in the estimation as shown in Figure 4.2. Here, the graph distribution obtained from the model in Figure 4.1 is used to infer a model, however, since each node of the second sample graphs is always mapped to the same model node, the resulting inferred model is different from the original one and it does not generate the same sample distribution.

To solve this bias Torsello [135] proposed to marginalize the sampling probability over all possible correspondences, which, once extended to deal with external nodes, results in the probability

$$P(\hat{g}|\mathcal{G}) = \sum_{\sigma\in\Sigma_n^m} P(g|\mathcal{G},\sigma)P(\sigma) = \frac{1}{|\Sigma_g|}\sum_{\sigma\in\Sigma_n^m} P(g|\mathcal{G},\sigma), \tag{4.1}$$

where $\hat{g}$ is is the quotient of $g$ modulo permutation of its nodes, i.e., the representation of $g$ where the actual order of the nodes is ignored, $\Sigma_n^m$ is the set of all possible partial correspondences between the $m$ nodes of graph $g$ and the $n$ nodes of model $\mathcal{G}$, and $\Sigma_g$ is the set of symmetries of $g$, i.e., the set of graph isomorphisms from $g$ onto itself.

Clearly, averaging over all possible correspondences is not possible due to the super-exponential growth of the size of $\Sigma_n^m$; hence, we have to resort to an estimation approach. In [135] was proposed an importance sampling approach to compute a fast-converging estimate of $P(g|\mathcal{G})$. Note that similar importance sampling approaches marginalizing over the space of correspondences have been used in [25] and [112]. In particular, in the latter work the authors show that the estimation has expected polynomial behavior.

### 4.1.1 Correspondence Sampler

In order to estimate $P(g|\mathcal{G})$, and to learn the graph model, we need to sample correspondences with probability close to the posterior $P(\sigma|g,\mathcal{G})$. Here we generalize the approach in [135] for models with external nodes, also eliminating the need to pad the observed graphs with dummy nodes to make them of the same size of the graph model.

Assume that we know the node-correspondence matrix $M = (m_{ih})$, which gives us the marginal probability that model node $i$ corresponds to graph node $h$. Note that, since model nodes can be deleted (not observed) and graph nodes can come from the external node model, we have that $\forall h, \sum_i m_{ih} \leq 1$ and $\forall i, \sum_h m_{ih} \leq 1$. We turn the inequalities into equalities by extending the matrix $M$ into a $(n+1)\times(m+1)$ matrix $\bar{M}$ adding $n+m$ slack variables, where the first $n$ elements of the last column are linked with the probabilities that a model node is not observed, the first $m$ elements of the last row are linked with the probability that an observed node is external and element at index $n+1, m+1$ is unused. $\bar{M}$ is a partial doubly-stochastic matrix, i.e., its first $n$ rows and its first $m$ columns add up to one.

With this marginal node-correspondence matrix to hand, we can sample a correspondence as follows: First we can sample the correspondence for model node 1 picking a node $h_1$ with probability $m_{1,h_1}$. Then, we to condition the node-correspondence matrix to the current match by taking into account the structural information between the sampled node and all the others. We do this by multiplying $\bar{m}_{j,k}$ by $P(g_{h_1,k}|\mathscr{G}_{1,j})$, i.e., the probability that the edges/non-edges between $k$ and $h_1$ map to the model edge $(1,j)$. The multiplied matrix is then projected to a double-stochastic matrix $\bar{M}_1^{h_1}$ using a Sinkhorn projection [130] adapted to partial doubly-stochastic matrix, where the alternate row and column normalization is performed only on the first $n$ rows and $m$ columns. We can then sample a correspondence for model node 2 according to the distribution of the second row of $M_1^{h_1}$ and compute the conditional matching probability $\bar{M}_{1,2}^{h_1,h_2}$ in much the same way we computed $M_1^{h_1}$. and iterate until we have sampled a complete set of correspondences, obtaining a fully deterministic conditional matching probability $\bar{M}_{1,\dots,n}^{h_1,\dots,h_n}$, corresponding to a correspondence $\sigma$, that has been sampled with probability $P(\sigma) = (\bar{M})_{1,h_1} \cdot (\bar{M}_1^{h_1})_{2,h_2} \cdot \ldots \cdot (\bar{M}_{1,\dots,n-1}^{h_1,\dots,h_{n-1}})_{n,h_n}$.

### 4.1.2 Estimating the Model

With the correspondence samples to hand, we can easily perform a maximum likelihood estimation of each model parameter by observing that, by construction of the model, conditioned on the correspondences the node and edge observation are independent to one another. Thus, we need only to maximize the node and edge models independently, ignoring what is going on in the rest of the graph. Thus, we define the sampled node and edge likelihood functions as

$$\begin{aligned}
\mathscr{L}_i(S,\mathscr{G}) &= \prod_{g\in S}\sum_{\sigma}\frac{P(g_{\sigma(i)}|\theta_i,\omega_i^n)}{P(\sigma)}\\
\mathscr{L}_{i,j}(S,\mathscr{G}) &= \prod_{g\in S}\sum_{\sigma}\frac{P(g_{\sigma(i),\sigma(j)}|\tau_{i,j},\omega_{i,j}^e)}{P(\sigma)}
\end{aligned}$$

from which we can easily obtain maximum likelihood estimates of the parameters $\theta_i$, $\omega_i^n$, $\tau_{i,j}$, and $\omega_{i,j}^e$.

Further, we can use th samples to update the initial node-correspondence matrix in the following way

$$\bar{M}' = \frac{1}{\sum_\sigma \frac{P(\sigma|g,\mathscr{G})}{P(\sigma)}}\sum_\sigma \frac{P(\sigma|g,\mathscr{G})}{P(\sigma)} M_\sigma$$

where $M_\sigma$ is the deterministic correspondence matrix associated with $\sigma$. Thus in our learning approach we start with a initial guess for the node-correspondence matrix and improve on it as we go along. In all our experiments we initialize the matrix based only on local node information, i.e. $m_{i,h}$ is equal the probability that model node $i$ generates the attributes of graph model $h$.

The only thing left to estimate is the value of $|\Sigma_g|$, but that can be easily obtained using our sampling approach observing that it is proportional to the probability of sampling an isomorphism between $g$ and a deterministic model obtained from $g$ by setting the values of $\tau_{i,j}$ to 1 or 0 according the existence of edge $(i,j)$ in $g$, and setting $\bar{\theta} = 0$. It interesting to note that in this corner case, our sampling approach turns out to be exactly the same sampling approach used in [18] to show that the graph isomorphism problem can be solved in polynomial time. Hence, our sampling approach is expected polynomial for deterministic model. and we can arguably be confident that it will perform similarly well for low entropy models.

### 4.1.3 Model Selection

Given this sampling machinery to perform maximum likelihood estimation of the model parameters for the naïve models, we adopt a standard EM approach to learn mixtures of naïve models.

This, however, leaves us with a model selection problem, since model likelihood decreases with the number of mixture components as well as with the size of the naïve models. To solve this problem we follow [136] in adopting a Minimum Message Length [143] approach to model selection, but we deviate from it in that we use the message length to prune an initially oversized model.

Thus we seek to minimize the combined cost of a two part message resulting in the penalty function

$$I_1 = \frac{D}{2}\log\left(\frac{|S|}{2\pi}\right) + \frac{1}{2}\log(\pi D) - 1 - \sum_{g\in S}\log\left(P(g|\mathcal{G},\sigma_g)\right), \tag{4.2}$$

where $|S|$ is the number of samples and $D$ the number of parameters for the structural model.

The learning process is initiated with a graph model that has several mixture components, each with more nodes that have been observed in any graph in the training set. We iteratively perform the EM learning procedure on the oversized model and, with the observation probabilities to hand, we decide whether to prune a node from a mixture component or a whole mixture component and after the model reduction we reiterate the EM parameter estimation and the pruning until no model simplification reduces the message length.

The pruning strategy adopted is a greedy one, selecting the operation that guarantees the largest reduction in message length given the current model parameters. Note that this greedy procedure does not guarantee optimality since the estimate is clearly a lower bound, as the optimum after the pruning can be in a very different point in the model-parameter space, but it does still give a good initialization for leaving the reduced parameter set.

In order to compute the reduction in message length incurred by removing a node, while sampling the correspondences we compute the matching probability not only of the current model, but also of the models obtained from the current one with any singe node removal. Note, however, that this does not increase the time complexity of the sampling approach and incurs only in a small penalty.

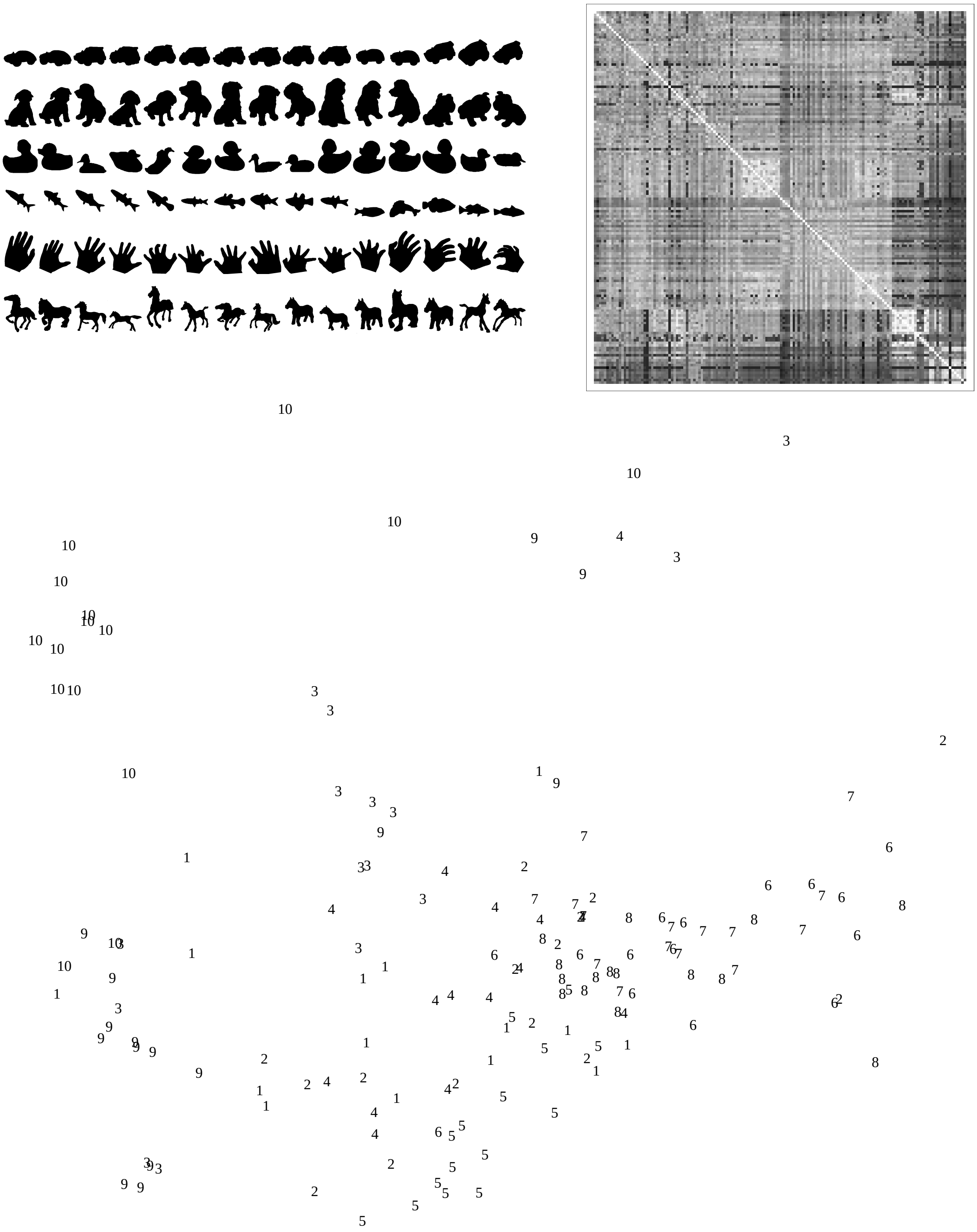


Figure 4.3: Top row: Left, a sample of the shape database; right, edit distance matrix. Bottom row: Multidimensional Scaling of the edit distances.

### 4.1.4 Experimental Evaluation

In order to asses the performance of the proposed approach, we run several experiments on graphs arising from different classification problems arising from 2D and 3D object recognition tasks, as well as one synthetic graph-classification testbed. The generative model is compared against standard nearest neighbor and nearest prototype classifiers based on the distances obtained using several graph matching techniques at the state of the art. In all cases the prototype is selected by taking the set-median of the training set. The performance of the generative model is assessed in terms of the classification performance for the classification task to hand. For this reason, for all the experiments we plot the precision and recall values:

$$\text{precision} = \frac{tp}{tp + fp} \qquad \text{recall} = \frac{tp}{tp + fn}$$

where $tp$ indicates the true positives, $tn$ the true negatives and $fn$ the false negatives.

With the exception to the last set of experiments, all the graphs used have a single numerical attribute associated to each node and no attributes linked with the edges. The last set of experiments, on the other hand, is based on edge-weighted graphs with no node attribute.

For the node-attributed graphs, we adopted the rectified Gaussian model used in [139]. To this end, we define a single stochastic node observation model $X_i$ for each node $i$. We assume $X_i$ is normally distributed with mean $\mu_i$ and standard deviation $\sigma_i$. When sampling node $i$ from the graph model, a sample $x_i$ is drawn from $X_i$. If $x_i \geq 0$ then the node is observed with weight $w_i = x_i$, otherwise the node will not be present in the sampled graph. Hence the node observation probability is $\theta_i = 1 - \text{erfc}(\mu_i/\sigma_i)$ where erfc is the complementary error function

$$\text{erfc} = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}s^2\right) ds.$$

The edge observation model, on the other hand is a simple Bernoulli process.

#### Shock Graphs

We experimented on learning models for shock graphs, a skeletal based representation of shape. We extracted graphs from a database composed of 150 shapes divided into 10 classes of 15 shapes each. Each graph had a node attribute that reflected the size of the boundary feature generating the corresponding skeletal segment. Our aim was to compare the classification results obtained learning a generative model to what can be obtained using standard graph matching techniques and a nearest neighbor classifier. Figure 4.3 shows the shape database, the matrix of extracted edit distances between the shock graphs, and a multidimensional scaling representation of the distances; here numbers correspond to classes. As we can see, recognition based on this representation is a hard problem, as the class structure is not very clear in these distances and there is considerable class overlap.

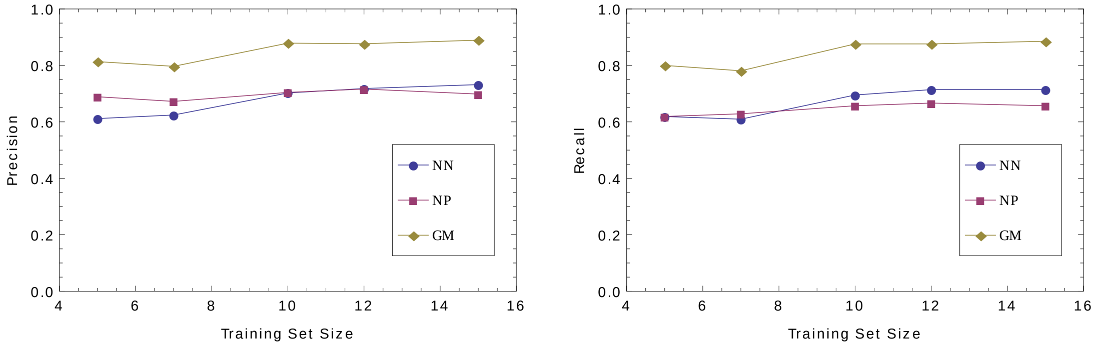


Figure 4.4: Precision and Recall on the shock graph dataset as the number of training samples increases.

In Figure 4.4 we compare the classification performance obtained with the nearest neighbor and nearest prototype rules with the one obtained by learning the generative models and using Bayes decision rule for classification, i.e., assigning each graph to the class of the model with largest probability of generating it. Note that the graphs are never classified with a model that had the same graph in the training set, thus in the case of the 15 training samples, the correct class had only 14 samples, resulting in a leave-one-out scheme. Figure 4.4 shows a clear improvement of about 15% on both precision and recall values regardless the number of samples in the training set, thus proving that learning the modes of structural variation present in a class rather than assuming an isotropic behavior with distance, as has been done for 40 years in structural pattern recognition, gives a clear advantage.

### 3D Shapes

The second test set is based on a 3D shape recognition task. We collected a number of shapes from the zhang2005retrieving 3D Shape Benchmark [152] and we extracted their medial surfaces using the algorithm introduced in Chapter 3. The final dataset was obtained by transforming these skeletal representations into an attributed graph. Figure 4.5 shows the shapes, their graph distance matrix and a Multidimensional Scaling representation of the distances. The distances between the graphs were computed using the normalized metric described in [140], which in turn relies on finding a maximal isomorphism between the graphs, for which we adopted the association graph-based approach presented in [109]. Both the distance matrix and the Multidimensional Scaling show that the classes are well separated, resulting in a relatively easy classification task.

Once again we tested the generative model performance against the nearest neighbor and the nearest prototype classifier. Figure 4.6 confirms our intuition that this was indeed an easy task, since both the nearest neighbor and the nearest prototype classifiers achieve the maximum performance. Yet, the generative model performs ex-

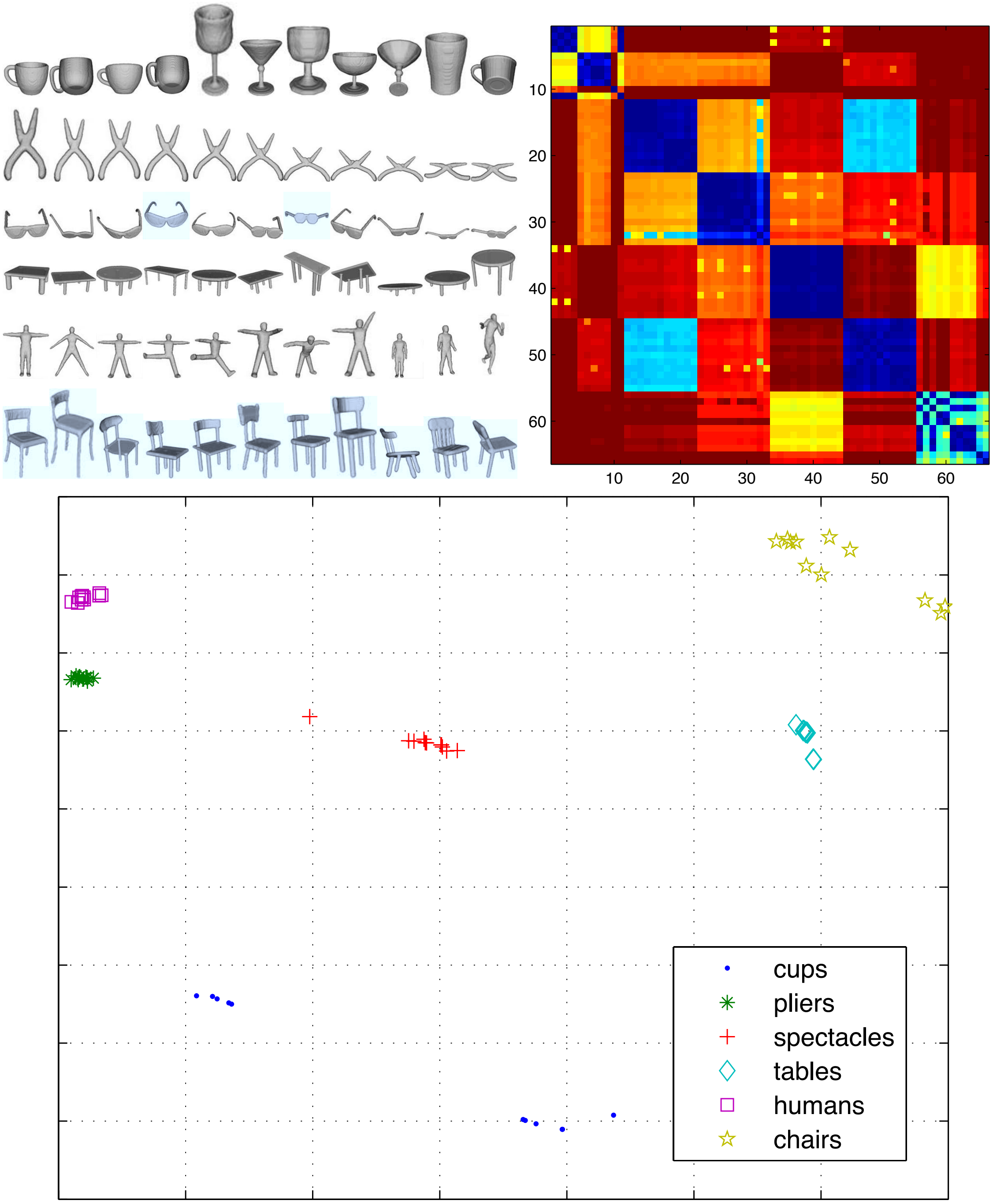


Figure 4.5: Top row: Left, shape database; right, distance matrix. Bottom row: Multidimensional Scaling of the graph distances.

tremely well, even when the training set contains just a very few samples. As for the performance gap between the nearest neighbor and the generative model, it is proba-

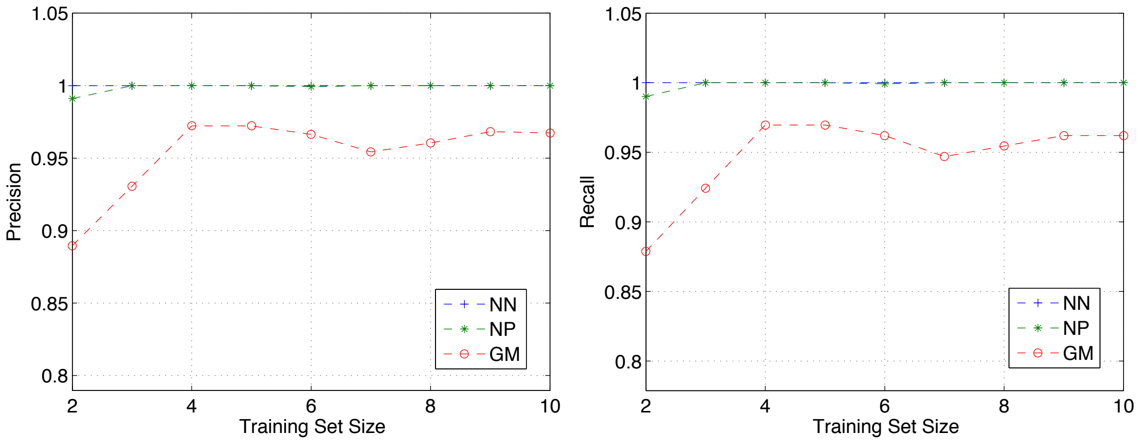


Figure 4.6: Precision and Recall on the 3D shapes dataset.

bly due to the very naïve way of estimating the initial node correspondences, and could be probably reduced using a more sophisticated initialization.

### Synthetic Data

To further assess the effectiveness of the proposed approach we tested it on synthetically generated data, where the data generation process is compatible with the naïve model adopted in the proposed learning approach. To this end, we have randomly generated 6 different weighted graph prototypes, with size ranging from 3 to 8 nodes. For each prototype we started with an empty graph and then we iteratively added the required number of nodes each labeled with a random mean and variance. Then we added the edges and their associated observation probabilities up to a given edge density. Given the prototypes, we sampled 15 observations from each class being careful to discard graphs that were disconnected. Then we proceeded as in the previous set of experiments computing the dissimilarities between the graphs and learning the graph models.

Generating the data with the same model used for learning might seem to give an unfair advantage to our generative model, but the goal of this set of experiments is asses the ability of the learning procedure to obtain a good model even in the presence of very large model-overlap. A positive result can also provide evidence for the validity of the optimization heuristics.

Figure 4.7 shows the distance matrix of the synthetic data and the corresponding Multidimensional Scaling representation. There is a considerable overlap between different classes, which renders the task particularly challenging for the nearest neighbor and nearest prototype classifiers. Yet, our generative model was able to learn and describe this large intra class variability, thus coping with the class overlap. Figure 4.8 plots the precision and recall curves for this set of experiments. Even with a relatively

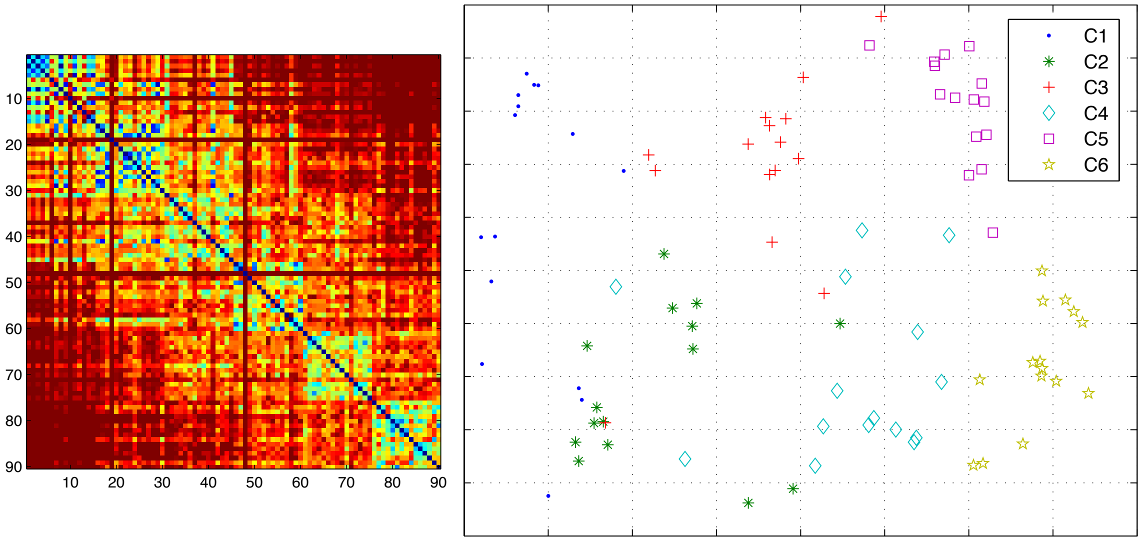


Figure 4.7: Distance matrix and MDS of distances for the Synthetic Dataset.

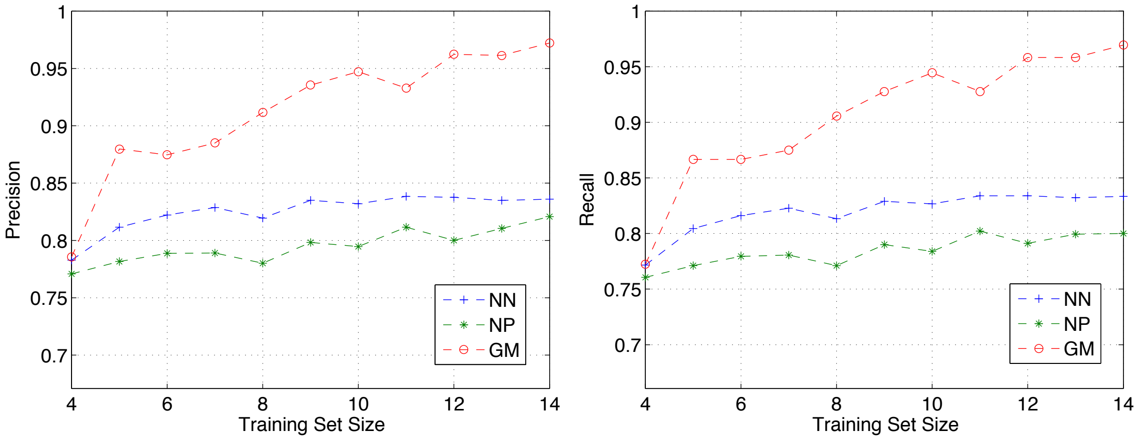


Figure 4.8: Precision and Recall on the synthetic dataset.

small training set, our approach achieves nearly 90% precision and recall, and as the number of observed samples increases, it yields perfect classification. On the other hand, the nearest neighbor classifier is not able to increase its precision and recall above the 84% limit, while the nearest prototype approach exhibits even lower performance.

### Edge-Weighted Graphs

In the finals set of experiments, we applied the approach to an object recognition task. To this end we used a subset of the COIL-20 dataset [101]. For each image we extracted the most salient points using a Matlab implementation of the corner detector described in [68], the salient points where connected according to a Delaunay triangulation, thus resulting in an edge-weighted graph, were the edge-weights correspond to the distance between the salient points.

With this representation we used different node and edge observation models. Since nodes are not attributed, we used simple Bernoulli models for them. For the edges, on the other hand, we used a combined Bernoulli and Gaussian model: a Bernoulli process establishes whether the edge is observed, and if it is the weight is drawn according to an independent Gaussian variable. The reason for this different weight model resides in the fact that the correlation between the weight and the observation probability that characterized the rectified Gaussian model did not fit the characteristics of this representation.

To compute the distances for the nearest neighbor and nearest prototype rule, we used the graph matching algorithm described in [46], which is capable of dealing with edge-weighted graphs. Once the correspondences where computed, we adopted the same metric as before. As Figure 4.9 shows, the generated dataset is even more complex than the synthetic one. This is mainly due to the instability of the corner detector, which provided several spurious nodes resulting in very large intra-class structural variability.

Figure 4.10 shows that even on this difficult dataset, we significantly outperform both the nearest neighbor and nearest prototype classifiers, emphasizing once again the advantages of our structural learning approach.

## 4.2 Information Theoretic Model Selection

We conclude this Chapter by introducing a novel approach to establish the optimality of a generative model. Standard model selection methods include the Minimum Message Length criterion (MML) [143], the Aikake [14] and the Bayesian information criteria [120]. In Section 4.1.3, for example, we have shown how to choose an optimal model according to a MML criterion. Generally speaking, although these principles are motivated from different viewpoints, most of them employ penalizing the parameters (or complexity) of the model in order to generalize well on a new dataset. For example, AIC assigns a cost to the model which is equal to twice the number of its parametres. Although these approaches have been used with considerable success for many years, they are all more or less sensible to the size of the training set. That is, if the observations are not enough these methods tend to underestimate the size of the generative model.

Here, on the other hand, we propose an alternative method which is specifically designed to establish the optimality of a model in terms of its generalization capabilities. The method we propose is an information-theoretic criterion that is inspired by the Approximation Set Coding framework [35]. The ASC framework has recently been introduced by Buhmann in the context of clustering validation, but our deviates from

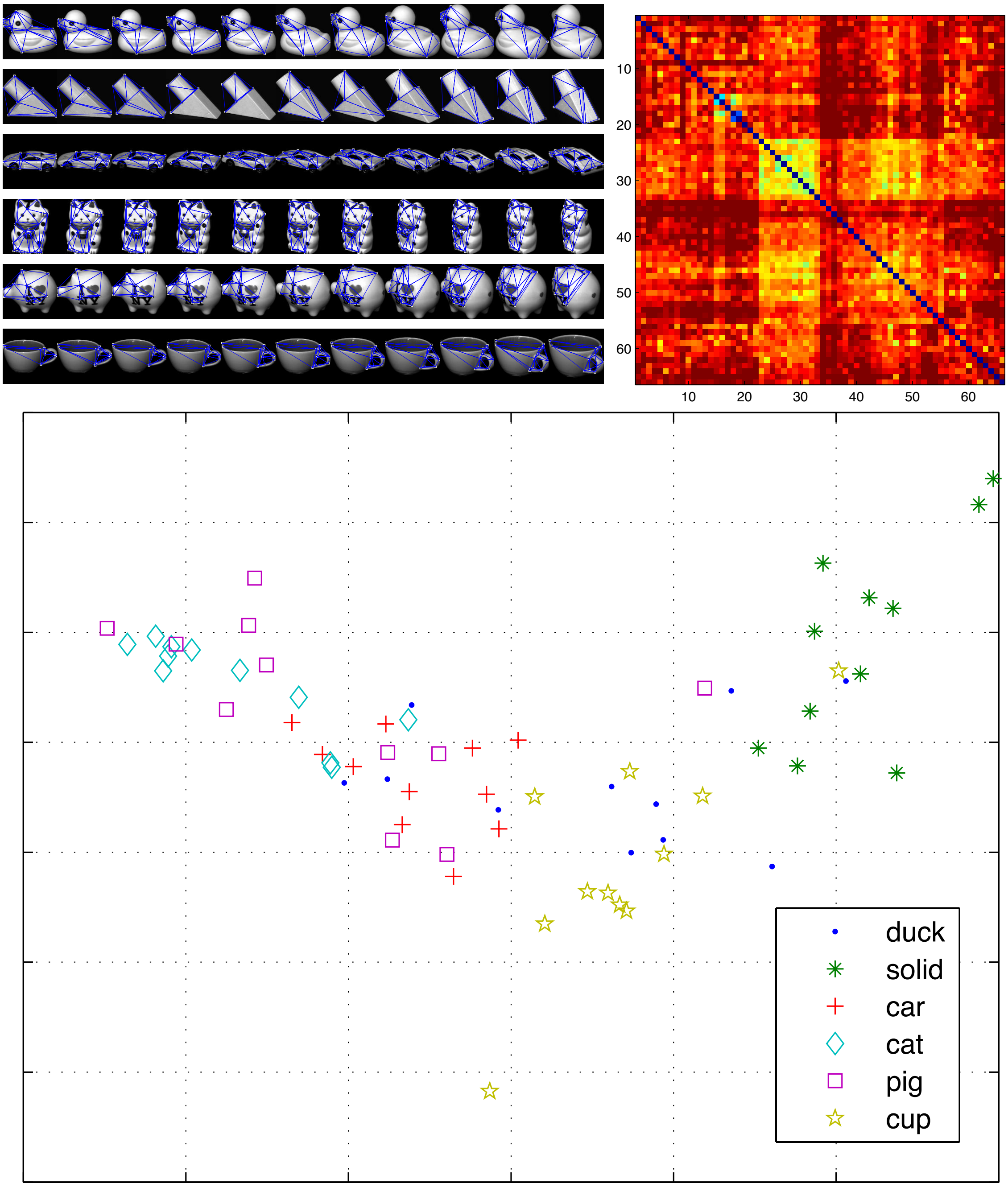


Figure 4.9: Top row: Left, shape database; right, distance matrix. Bottom row: Multidimensional Scaling of the graph distances.

it under several aspects. We first review the ASC framework of Buhmann and then we proceed to explain our modified version.

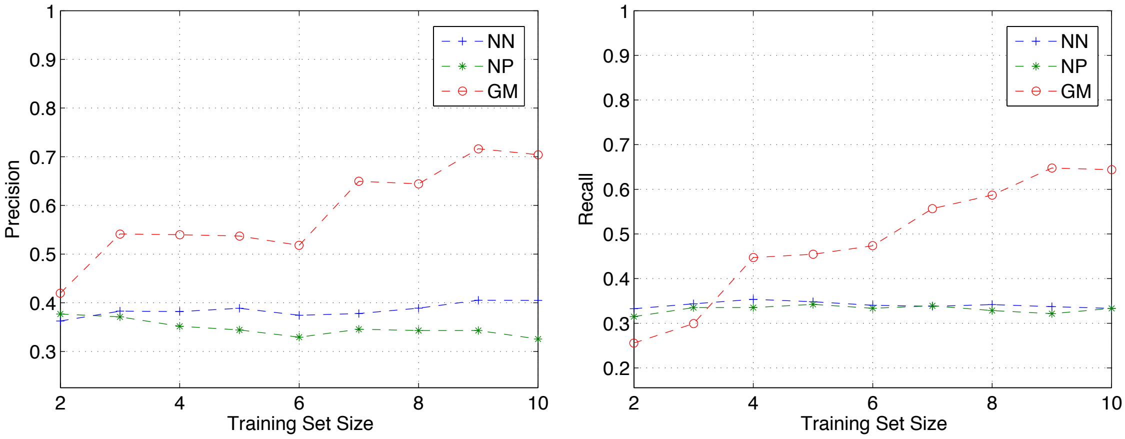


Figure 4.10: Precision and Recall on the COIL-20 dataset.

### 4.2.1 Approximation Set Coding

We are given a set of objects $\mathbf{O} = \{o_1, ..., o_n\}$ which is characterized by a set of measurements $\mathbf{X}$ associated with these objects. Let a *hypothesis* be a solution to our pattern recognition problem. In particular, in [35] a hypothesis $c$ is defined as a function assigning data to clusters. Let $R(c)$ be the *risk function* associated with a particular clustering algorithm, i.e., a function which evaluates the quality of a hypothesis according to a specific criterion of coherency. In the case of k-means, the risk function measures the average distance of the objects to the nearest cluster centroid. Given a set of measurements and of alternative clustering solutions, the best hypothesis $c^\perp$ is defined as the hypothesis that minimizes the empirical risk of data clustering given the measurements $\mathbf{X}$, i.e.,

$$c^\perp(\mathbf{X}) = \arg\min_c R(c, \mathbf{X}) \tag{4.3}$$

Then the set $C_\gamma(\mathbf{X})$ of empirical risk approximations for clustering is defined as

$$C_\gamma(\mathbf{X}) := \{c(\mathbf{X}) : R(c, \mathbf{X}) \leq R(c^\perp, \mathbf{X}) + \gamma\} \tag{4.4}$$

that is the set of hypotheses that are at most $\gamma$-far from the optimal one, in terms of the risk function.

Given this setting, a clustering algorithm $A$ should be validated by evaluating the generalization properties of the clustering solutions $c(\mathbf{X}^{(1)})$ on a set of test data with associated measurements $\mathbf{X}^{(2)}$, where $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ represent the training and test sets respectively. Therefore, one needs to define a mapping $\psi$ which identifies a hypothesis for the training set with a hypothesis for the test data. With this mapping to hand, it is then possible to enumerate the $\gamma$-optimal clustering solutions which are also $\gamma$-optimal on the test data. Note that here $\gamma$ controls the trade off between the generalization power and the informativeness of the clustering algorithm.

Approximation set coding uses the observation that there are a set of problem specific invariants, i.e., a set of transformations which alter the sample data without essentially changing the clustering model in any way. In the clustering problem, these are the random permutations $\Sigma = \{\sigma_1, ..., \sigma_{2^{nR}}\}$ of the objects. The generalization property of a clustering algorithm is then evaluated in the following noisy communication scenario. Initially, a sender S and a receiver R agree on a clustering algorithm with an associated risk function $R(c, X^{(1)})$ and a mapping function $\psi$. Then, $S$ and $R$ obtain the data $\mathbf{X}^{(1)}$ from the problem generator PR, which also generates the set of random permutations $\Sigma$. With $\mathbf{X_1}$ and $\Sigma$ to hand, both S and R determine the approximation sets $C_\gamma(\sigma_i \circ \mathbf{X}^{(1)})$, $1 \leq i \leq 2^{n^R}$. In this communication scenario, the approximation sets $C(\sigma_i \circ \mathbf{X}^{(1)})$ play the role of Shannon's codebook vectors. Finally, the communication takes place in the following way: 1) the sender S selects the permutation $\sigma_s$ and sends it to the problem generator PR; 2) PR generates a new dataset $\mathbf{X}^{(2)}$ and its permuted version $\tilde{\mathbf{X}} = \sigma_s \circ \mathbf{X}^{(2)}$; 3) PR sends $\tilde{\mathbf{X}}$ to R, which then computes the approximation set $C_\gamma(\tilde{\mathbf{X}})$; 5) finally, R estimates the applied permutation as

$$\tilde{\sigma} = \arg\max_{\sigma \in \Sigma} \left| \left( \psi \circ C_\gamma(\sigma \circ \mathbf{X}^{(1)}) \right) \cap C_\gamma(\tilde{\mathbf{X}}) \right| \tag{4.5}$$

With this setting to hand, one can select the optimal clustering method as the one which yields the highest channel capacity in the transmission of the invariant $\sigma_s$. Note that Buhmann's framework is designed to validate a clustering algorithm. As a consequence, in order to apply it to our model selection problem we need to understand the fundamental distinction between validating a meta-model (or algorithm), as opposed to simply validating a model, which is a lower level problem.

### 4.2.2 Approximation Set Coding For Model Selection

Han et. al [65] recently proposed to extend Buhmann's framework from the vectorial domain to the graph domain in order to solve a prototype size selection problem. However, the authors are actually validating a learning algorithm, rather than a graph prototype. In our work, we intend to correct their analysis by modifying in a simple yet fundamental way the communication scenario.

In the graph model validation setting, a hypothesis $c$ is a mapping (match) of the sample graphs to a model graph. As in the usual ASC framework, we have a risk function $R(c)$ which evaluates the cost of a particular matching. In order to evaluate the generalization properties of the model, we need to be able to transfer the matches from the training set of graphs $\mathbf{G}^{(1)}$, to the test set $\mathbf{G}^{(2)}$. For each graph $g_i^{(1)}$ in $\mathbf{G}_1$, we find the most similar graph in $\mathbf{G}_2$ and the mapping between $T_i$ between the two, where the similarity is computed in terms of graph edit distance. Thus $T_i \circ g_i^{(1)}$ is the image of $g_i^{(1)}$ in the second set.

In this new setting, the invariants that we need to transmit are the permutation of the sample graphs and of their nodes. Note, in fact, that if we consider the sample graphs in a different order, or their nodes are permuted in some way, the structure of the recovered model should be the same (although the model graph nodes may also be in a different order). However, the new communication scenario requires an important modification of in the initial setup of the channel. In this case, the sender S and the receiver R need to agree also on the model $\mathcal{G}$ which will be used to compute the approximation sets, i.e., the codebook vectors. In other words, while in [65] at the two ends of the channel S and R rely on two different models to compute the approximation sets, which implies that a meta-model rather than a model validation is taking place, in our setting S and R need to operate given the same model which is initially sent to both by the problem generator PR.

Then, given a set of models $\{\mathcal{G}_i\}_{i=1}^n$, we select the model which yields the highest channel capacity, i.e., the one which maximizes the mutual information between S and R. As shown by [35], the mutual information between sender and receiver can be computed as

$$I_\gamma = \frac{1}{n}\log\left(\frac{|T||\Delta C_{\gamma,12}|}{|C_{\gamma,1}||C_{\gamma,2}|}\right) \tag{4.6}$$

where $|C_{\gamma,1}|$ is the number of hypotheses which are within a cost $\gamma$ of the best cost in set 1 (and likewise for $|C_{\gamma,2}|$). The quantity $|\Delta C_{\gamma,12}|$ is the number of hypotheses in set 2 which are within a cost $\gamma$ of the best cost in set 1, where we need to define a way of transferring hypotheses from set 2 to set 1. More precisely, unlike [65], for each model we compute the average mutual information by repeatedly partitioning the set of observations into $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$, thus reducing the dependence of the chosen model from the quality of the partition. Clearly this is not possible in [65], as the models used to encode and decode the invariants are learned given the partitioned observations.

### 4.2.3 Model Selection Framework

In this Section, we show how to extend the methodology of the approximate set coding from the vector domain to the graph domain. To this end, we redefine three important ingredients in the approximate set coding (i.e. hypothesis, cost function and partition function), and generalize them from vector domain to graph domain. In the following, we commerce by introducing our problem and then explain the new definition of the ingredients.

Given a set of sample graphs, our aim is to select the optimal model graph for the sample graphs. To ensure that the optimal model graph generalizes well on new dataset, we adopt a two-sample set scenario and repeatedly partition the sample graphs into two sets of the same size

$$\begin{aligned}\mathbf{G}^{(1)} &= \{g_1^{(1)}, g_2^{(1)}, \ldots, g_n^{(1)}\}\\ \mathbf{G}^{(2)} &= \{g_1^{(2)}, g_2^{(2)}, \ldots, g_n^{(2)}\}\end{aligned} \tag{4.7}$$

Here the superscripts indicate different sample-set and the subscripts indicate the graph indices. The best model graph is determined according to its average generalization capability on the two sets.

### Hypothesis And Cost Function

The hypotheses originally proposed in the clustering problem are the assignments of data points to clusters [34]. In this work, the hypotheses consist of the set of correspondences of each of the sample graphs onto its corresponding model graph. By direct analogy with the clustering problem, each correspondence is equivalent to an assignment of a point to a cluster; the model graph here is a parameter equivalent to the cluster centroid. For each dataset $\mathbf{G}^{(q)} (q \in \{1,2\})$ a hypothesis is

$$c_q = \{\sigma_1^{(q)}, \sigma_2^{(q)}, ..., \sigma_n^{(q)}\} \tag{4.8}$$

where $\sigma_i^{(q)}$ $(i \in \{1,...,n\})$ is the assignment between graph $i$ from set $q$ and the model graph $\mathcal{G}$. The set of all possible hypotheses is $\Sigma$ and it consists of all the possible correspondences between all samples and the model graph.

Furthermore, we need a cost function $R_q(c_q)$ to quantify the effectiveness of a particular hypothesis $c_q$. The cost function measures how consistent the given correspondences are with the model graph. Here the cost function of a hypothesis is the negative logarithm of the joint observation probability of the graphs $g_i^{(q)} \in \mathbf{G}^{(q)}$

$$\begin{aligned} R_q(c_q) &= -\log P(\mathbf{G}^{(q)}|\mathcal{G}, c_q) \\ &= \sum_{g_i^{(q)}} \Bigg( (1-\bar{\theta}) + \sum_{v \in V} P\Big(g_i^{(q)}(\xi_i(v))|\theta_v\Big) + \sum_{v \notin V} P\Big(g_i^{(q)}(\xi_i(v))|\bar{\theta}_v\Big) + \\ &\sum_{(u,v) \in E} P\Big(g_i^{(q)}(\xi_i(u), \xi_i(v))|\tau_{u,v}\Big) + \sum_{(u,v) \notin E} P\Big(g_i^{(q)}(\xi_i(u), \xi_i(v))|\bar{\tau}_{u,v}\Big)\Bigg) \end{aligned} \tag{4.9}$$

where $\xi = \sigma^{-1}$. In order to normalize the minimum cost of the hypotheses to zero, we define the relative cost of hypothesis. Suppose the optimal hypothesis (i.e., the hypothesis yielding the lowest costs between the sample graphs and the model graph) is $c_q^{\perp}$, the relative cost of hypothesis $c_q$ is

$$\Delta R_q(c_q) = R_q(c_q) - R_q(c_q^{\perp}) \tag{4.10}$$

### Partition Function

The measurement of the mutual information of the two sample-set requires counting the number of the hypotheses which are within a certain cost of the optimal solution. However, this is hard to do as it involves exploring all the hypotheses. Fortunately, this value can be estimated using some concept from statistical physics. Considering the hypotheses as microcanonical ensembles in statistical mechanics, their number can be estimated by calculating the partition function [34]

$$\mathcal{Z}_q = \sum_{c_q \in \mathcal{C}_q} \exp[-\beta \Delta R_q(c_q)] \tag{4.11}$$

where $\beta$ is a positive scaling parameter known as the inverse computational temperature. Essentially, $\beta$ coarsens the precision of the partition function approximating the number of hypotheses that fit the sample set [35]. When $\beta$ is zero, the partition function is equal to the number of all the possible hypotheses. When $\beta$ is very large, the partition function only counts the number of optimal hypotheses. Because $\beta$ controls the number of hypotheses fitting the sample set, we will call these $\beta$-optimal hypotheses. In our case, the hypotheses space is the set of all the possible correspondences between the sample graphs and the model graph. The hypotheses space is very large and the computation of the partition function will be expensive. Later we show how we use the Importance Sampling approach to sample the correspondence between the sample graphs and the model graph and approximate the partition function.

To measure how well the hypotheses generalize for the two sample sets, we count the number of $\beta$-optimal hypotheses in the first set which also exist in the second set, when transferred to the first set. We therefore need a way of transferring hypotheses from the second dataset to the first. We denote the cost of the hypothesis $c_2$ between the transferred graphs and model graph $\mathscr{G}$ as $R_t(c_2)$. This is the cost of making hypothesis $c_2$ for the graphs $\mathbf{G}^{(2)}$ when evaluated against the data in $\mathbf{G}^{(1)}$. The following procedure may be used to find the transfer. For each $g_i^{(1)}$ graph in $\mathbf{G}^{(1)}$, we find the most similar graph in $\mathbf{G}^{(2)}$ and the correspondence between $T_i$ between the two. $T_i \circ g_i^{(1)}$ is then the image of this graph in the second set. From these images, we compute the cost of $c_2$ by comparing the images to the model graph $\mathscr{G}$ under the correspondences in $c_2$. Finally, the joint partition function is formulated as

$$\mathscr{Z}_{12} = \sum_{c_2 \in \mathscr{C}_2} \exp[-\beta(\Delta R_t(c_2) + \Delta R_2(c_2))] \,. \tag{4.12}$$

The quantity $\Delta R_t(c_2)$ is the relative cost of hypothesis $c_2$ between the image graphs of $\mathbf{G}^{(1)}$ in the second set and the model graph $\mathscr{G}$. It is equivalent to the cost of hypothesis $c_2$ between the image graphs and $\mathscr{G}$ minus their minimum cost.

The model graphs can then be ranked according to their mutual information between the two sets

$$I_\beta = \frac{1}{n}\log\left(\frac{Z_{12}}{Z_1 Z_2}\right) \tag{4.13}$$

In the above equation, $Z_1$,$Z_2$ are the respective partition functions of two sample sets and $Z_{12}$ is their joint partition function. Note, however, that $I_\beta$ depends on the given partition of the data. Hence, we propose to repeatedly partition the data and compute the average mutual information between the pair of sets, rather then the mutual information.

### Partition Function Approximation

In order to deal with the super-exponential growth of the set of possible correspondences, we decide to resort to an Importance Sampling approach in a manner which is similar to that of Section 4.1.1. Importance Sampling is a variance reduction sampling technique used when computing Monte Carlo approximations [64]. Suppose we want to estimate the average $E[h(x)] = \frac{1}{||A||}\int_A h(x)dx$, where $h(x)$ is a real function taking values in $A$. If we sample $k$ values of $h(x)$ uniformly we obtain the Monte Carlo estimator $E[h(x)] \approx \frac{1}{k}\sum_{i=1}^{k} h(x_i)$. However, it is often the case that in some of the regions of $A$ the value of $h(x)$ is very small, i.e. its impact to the estimate is negligible. One way to overcome this problem would be that of taking a larger number $k$ of samples, but the reason why we resorted to a Monte Carlo approximation in the first place was actually to avoid enumerating all the possible correspondences. If instead we sample from a different and non necessarily uniform distribution $f$, we can estimate $E[h(x)]$ as

$$E_f[h(x)] \approx \frac{1}{k}\sum_{i=1}^{k} h(x_i)\frac{\frac{1}{||A||}}{f(x_i)} \tag{4.14}$$

where $\frac{\frac{1}{||A||}}{f(x_i)}$ is called the *importance factor.* In other words, we are taking a weighted sum of the $h(x_i)$, where the importance factor is used to correct the error introduced when sampling from the new distribution $f$. Note that if we choose $f(x) = \frac{h(x)}{\int_A h(x)dx}$, the variance of the estimator is zero and thus a single sample is sufficient to estimate $E[h(x)]$. However this would require computing $\int_A h(x)dx$, so in practice one should choose $f$ to be as close as possible to $\frac{h(x)}{\int_A h(x)dx}$.

In this work, we need to approximate the value of the partition functions $\mathcal{Z}_1$, $\mathcal{Z}_2$ and $\mathcal{Z}_{12}$. Since the approximation procedure is going to be the same in all the three cases, we simply review the equations for $\mathcal{Z}_1$. In this case, $||A|| = n!$ and $h(x) = \exp[-\beta\Delta R_1(c_1)]$, and thus

$$\mathcal{Z}_1 = E_{c_1}\Big[\exp[-\beta\Delta R_1(c_1)]\Big]n! \approx \frac{1}{|\mathscr{C}_1|}\sum_{c_1\in\mathscr{C}_1}^{k}\frac{\exp[-\beta\Delta R_1(c_1)]}{P(c_1)} \tag{4.15}$$

In order to implement the importance sampler we follow the approach of Section 4.1.1. Recall that $\Delta R_q = R_q(c_q) - R_q(c_q^{\perp})$ and $R_q(c_q) = -\log P(\mathbf{G}^{(q)}|\mathscr{G}, c_q)$, where $\mathbf{G}^{(q)}$ is a set of observations and $\mathscr{G}$ is the model graph. Hence, for each graph $g_i^{(q)} \in \mathbf{G}^{(q)}$, we want to sample a correspondence $\sigma_i^{(q)}$ with probability close to

$$\frac{P(g_i^{(q)}|\mathscr{G})}{\sum_{\sigma_i^{(q)}} P(g_i^{(q)}|\mathscr{G}, \sigma_i^{(q)})} \tag{4.16}$$

The sampling of the correspondences then follows closely that of Section 4.1.1. Note, however, that the procedure described in this Section actually aims at providing a good approximation of the first moment of $\exp[\Delta R_q(c_q)]$, rather than its $\beta$th moment, as one would expect. In fact, the correct solution would require repeating the sampling procedure for each desired $\beta$, which is clearly unfeasible. However, in the experimental part we will show that even with this simplification the resulting approximation is still satisfying.

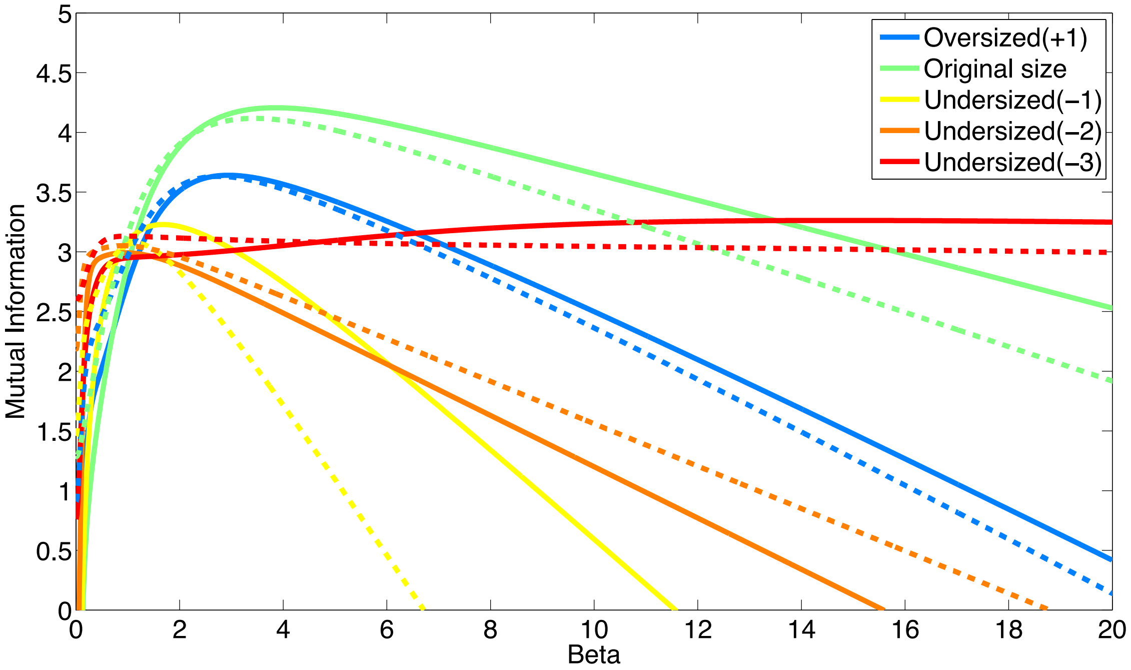


Figure 4.11: The variation of the mutual information as the inverse temperature $\beta$ increases. The solid and the dotted lines show the exact and the approximated values of the mutual information, respectively.

### 4.2.4 Experimental Evaluation

The novelty of the proposed model selection framework requires an extensive and careful experimental validation both on synthetic and real-world graphs. In particular, in this Section we study the quality of the importance sampling approximation and we compare our method with several standard model selection criteria.

#### Synthetic Data

To begin with, we evaluate our approach on a set of synthetically generated graphs. A total of 50 observations are sampled from a generative model with 6 nodes, where the node and edge observation probabilities are set randomly. Hence, we know the original model and, most importantly, we are aware of the existence of an underlying generative model which can reasonably describe our observations. On the other hand, in many real-world datasets the heteroscedasticity of the observations renders learning a model extremely hard, if not impossible.

We start by studying the error caused by the importance sampling approximation described in Section 4.2.3. To this end, we need to compute the exact value of the partition functions, which motivates our choice of a relatively small size for the generative model. Given the set of 50 observations, a total of 5 models are learned, with sizes ranging from 7 to 3. Hence, we learn one oversized model, one model with ground

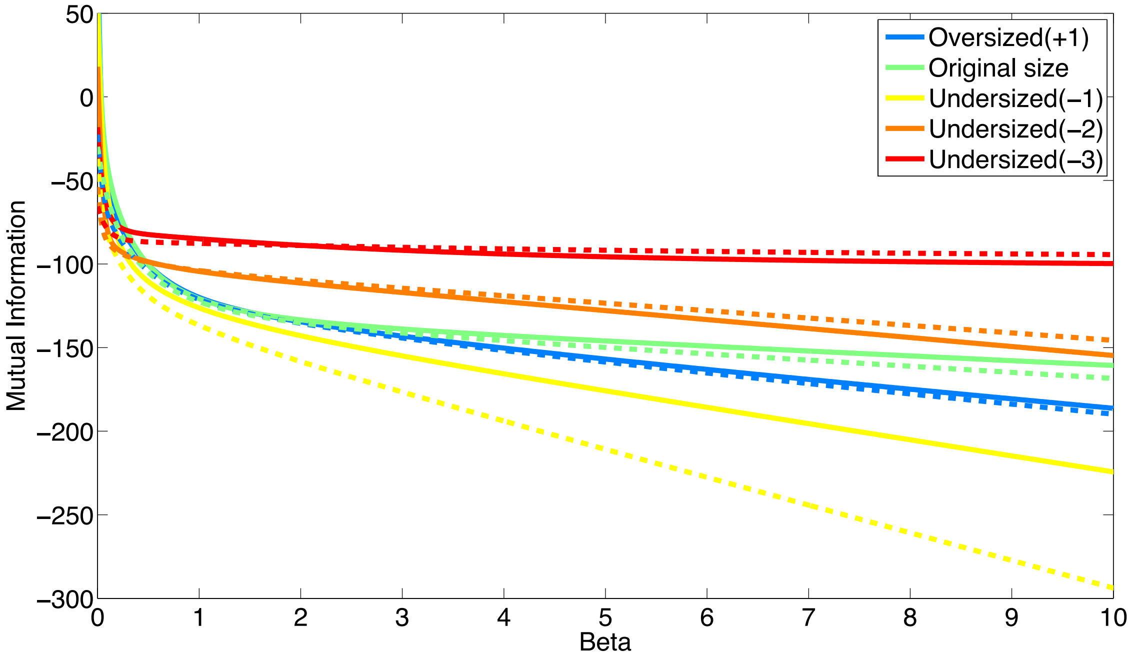
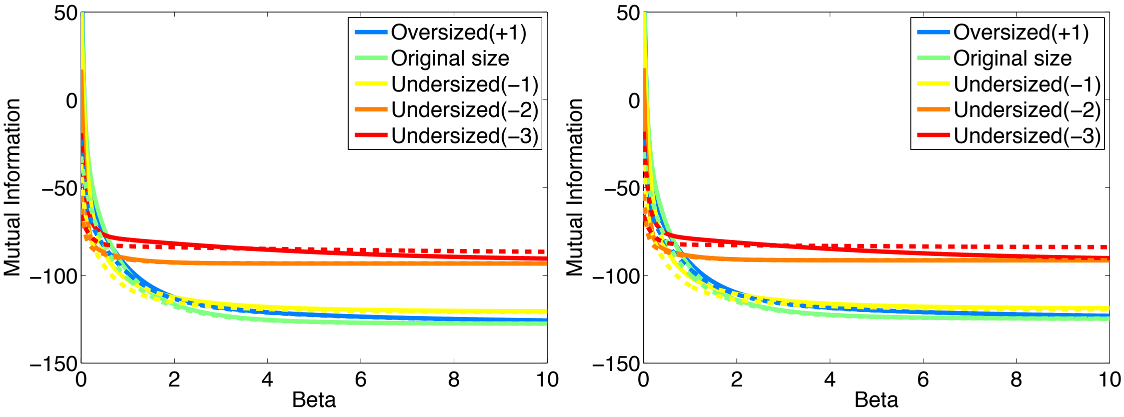


Figure 4.12: The variation of $\log \mathcal{Z}_{12}$ as the inverse temperature $\beta$ increases. The solid and the dotted lines show the exact and the approximated values of the mutual information, respectively.

truth size, and 3 undersized model. The reason why we learn only one oversized model is that every oversized instance converged to a model with no more than 6 nodes, that is the observation probability of the nodes dove to 0 on all nodes except 6.

Figure 4.11 shows the values of the mutual information as the inverse temperature $\beta$ increases. Here the solid lines indicate the values of the exact functions, while the dotted lines indicate the values of the approximated functions. We first note that, independently of the importance sampling approximation, the model yielding the highest mutual information is the one which has the same size of the original model. However, we observe a misestimation of the mutual information when the importance sampling is introduced. The reason lies in the relatively poor approximation of $\mathcal{Z}_{12}$, as shown in Figure 4.12. In fact, the correspondence sampler described in Section 4.2.3 is designed to give a good approximation of $\mathcal{Z}_1$ and $\mathcal{Z}_2$, rather than $\mathcal{Z}_{12}$. This is clear from Figure 4.13(a) and 4.13(b), which shows a close to perfect match between the approximated and the exact values of $\log \mathcal{Z}_1$ and $\log \mathcal{Z}_2$. If we go back to Figure 4.12, we can observe that the gentler the slope of $\log \mathcal{Z}_{12}$, the easier it is to transfer a correspondence from $\mathbf{G}_1$ to $\mathbf{G}_2$. In fact, when the inverse temperature increases, $\exp[-\beta(\Delta R_t(c_2)+\Delta R_2(c_2))]$ is dominated by the correspondences $c_2$ for which $\Delta R_t(c_2)+\Delta R_2(c_2)$ is close to zero, i.e. those correspondences which are optimal both in $\mathbf{G}_1$ and in $\mathbf{G}_2$. Note also that we do not want our correspondence sampler to pick always the optimal correspondence between a graph and a model, as this would underestimate the difficulty



(a) $\log \mathscr{Z}_1$

(b) $\log \mathscr{Z}_2$

Figure 4.13: The variation of $\log \mathscr{Z}_1$ and $\log \mathscr{Z}_2$ as the inverse temperature $\beta$ increases. The solid and the dotted lines show the exact and the approximated values of the mutual information, respectively.

of the transfer, i.e. the value of the approximated $\log \mathscr{Z}_{12}$ would be higher than in the exact case. This is for example the case of the second undersized model.

We should stress, however, that the errors introduced by the importance sampling approximation do not prevent us from being able to chose the correct model as the one which maximizes the mutual information. More precisely, the relative order between the different models seems to be unaffected by the approximation. On the other hand, note that we have a great divergence between the real and the approximated values when $\beta$ tends to zero. In particular, when $\beta = 0$ the value of $\mathscr{Z}_1$ should be equal to the number of possible correspondences between the sample graphs and the models, while in the importance sampling approach we get $\mathscr{Z}_1 \approx \frac{1}{|\mathscr{C}_1|} \sum_{c_1 \in \mathscr{C}_1}^{k} \frac{\exp[-\beta \Delta R_1(c_1)]}{P(c_1)} = \sum_{c_1 \in \mathscr{C}_1}^{k} P(c_1)^{-1}$, which is clearly wrong. Recall, that this is a consequence of the fact that we are approximating the first moment rather than the $\beta$th moment of $\exp[\Delta R_1(c_1)]$. As we observed, however, this simplification, which is required for the framework to be feasible, still yields reasonable results.

Finally, Figure 4.13(a) and Figure 4.13(b) show that the values of $\log \mathscr{Z}_1$ and $\log \mathscr{Z}_2$ converge to a horizontal asymptote. In fact, as $\beta$ increases, $\log \mathscr{Z}_q$ is essentially counting the number of optimal hypothesis, i.e., the correspondences for which $\Delta R_q(c_q) = 0$. On the other hand, Figure 4.12 shows that the optimal correspondence of the graphs in the test set do not necessarily generalize to the optimal correspondence of their mapped graphs in the training set, and hence $\log \mathscr{Z}_{12}$ does not converge to a horizontal asymptote. However, as already noted the steepness of the slope depends on the difficulty of transferring the hypotheses between the two sets. Clearly, the smaller the model the easier it is to transfer a hypothesis, as the external node will have a higher observation probability. In fact, the external nodes increases the number of symmetries of the model and thus the number of optimal graph-to-model correspondences.

| Dataset | AIC | BIC | MML | MI |
|---|---|---|---|---|
| MUTAG (overall) | 0.632 | 0.632 | 0.632 | **0.684** |
| *MUTAG (class 1)* | 0.720 | 0.720 | 0.720 | 0.600 |
| *MUTAG (class 2)* | 0.462 | 0.462 | 0.462 | 0.847 |
| Letters (overall) | **0.732** | 0.720 | 0.720 | 0.720 |
| *Letters (A)* | 0.793 | 0.793 | 0.793 | 0.828 |
| *Letters (E)* | 0.931 | 0.931 | 0.931 | 0.931 |
| *Letters (H)* | 0.965 | 0.965 | 0.965 | 0.965 |
| *Letters (M)* | 0.690 | 0.621 | 0.621 | 0.621 |
| *Letters (N)* | 0.655 | 0.655 | 0.655 | 0.655 |
| *Letters (V)* | 0.758 | 0.758 | 0.758 | 0.758 |
| *Letters (W)* | 0.207 | 0.241 | 0.241 | 0.207 |
| *Letters (X)* | 0.665 | 0.665 | 0.665 | 0.665 |
| *Letters (Y)* | 0.931 | 0.931 | 0.931 | 0.931 |

Table 4.1: The classification accuracy on the MUTAG and Letters datasets for different choices of the model selection method. Here MML denotes the Minimum Message Length criterion, AIC denotes the Aikake Information Criterion, BIC denotes the Bayesian Information Criterion and finally MI indicates the proposed Mutual Information criterion. Note that for each dataset the best accuracy is highlighted in bold.

However, for the same reason such a model will also a higher limiting value of $\log \mathcal{Z}_1$ and $\log \mathcal{Z}_2$, which will penalize the value of its mutual information $I = \log \mathcal{Z}_{12} - \log \mathcal{Z}_1 - \log \mathcal{Z}_2$.

**Real-World Data**

Our aim here is that of comparing the proposed model selection method to standard methods which are commonly used in the literature, namely the Minimum Message Length criterion (MML) [143], the Aikake [14] and the Bayesian information criteria [120]. Here we evaluate the goodness of a model selection method in terms of classification accuracy. More precisely, given a dataset of graphs we partition each it into training data and testing data. We then learn a set of generative models of different sizes for each class. The goodness of a model selection method is thus evaluated as the classification accuracy achieved by the optimal model, according to that method.

The experiments are performed on two real-world dataset, namely MUTAG and the Letters dataset from the IAM Graph Database Repository [116]. MUTAG is a dataset of 188 mutagenic aromatic and heteroaromatic compounds labeled according to whether or not they have a mutagenic effect on the Gram-negative bacterium *Salmonella typhimurium*. The Letters dataset consists of 15 capital letters from the Roman alphabet. These letters are chosen so that they can be drawn using only straight lines, which makes it easy to map a letter to a graph. More precisely, the edges of the graph correspond to the straight lines, whose endpoints are encoded as the graph nodes. Note that in the original dataset each node is labeled with a two-dimensional attribute giving the coordinates the corresponding endpoint, while the edges are unlabeled. In our experiments, however, we will drop the node attributes, and as a consequence we need to prune the dataset to avoid including letters which share the same structure, e.g., N and Z, or A and K.

Table 4.1 shows the results of the experiments in terms of classification accuracy. In particular, for each model selection method, we show the overall accuracy along with the per class accuracy. Note that the proposed model yields the highest classification accuracy in the MUTAG dataset, while in the Letter dataset its performance is comparable to that of MML and BIC.

## 4.3 Conclusions

In this Chapter we have addressed the problem of learning a generative model for graphs from samples. The model is based on a naïve node independence assumptions, but mixes such simple models in order to capture node correlation. The correspondences are estimated using a fast sampling approach, the node and edge parameters are then learned using maximum likelihood estimates, while model selection adopts a minimum descriptor length principle. Experiments performed on a wide range of real world object recognition tasks as well as on synthetic data show that learning the graph structure gives a clear advantage over the isotropic behavior assumed by the vast majority of the approaches in the structural pattern recognition literature. In particular, the approach very clearly outperforms both the nearest neighbor and the nearest prototype rules regardless of the matching algorithm and the distance metric adopted.

Moreover, in this Chapter we have introduced a novel information-theoretic method for model selection. The optimal model is selected so as to maximize the mutual information of the two partitioned sets of observed graphs. To compute the mutual information, we extended the theory of approximate set coding from the vector domain to the graph domain. Experimental results showed that our model selection criterion performs comparably to other widely used methods such as MML, AIC and BIC, beating all of these methods in a bioinformatic dataset.

In the next Chapter we will introduce turn our attention from generative to discriminative classification approaches. Discriminative approaches are generally characterized by a higher classification performance, but they usually work in a vectorial space. However, it is not clear how to embed a graph onto a vectorial space. Using kernel methods, rather then explicitly introducing a vectorial space, one simply needs to define the kernel measure between two graphs, which, if certain conditions are satisfied, will correspond to a dot product in an implicitly defined vectorial space. The next Chapter will be then dedicated to the introduction of a novel graph kernel.

# 5

# Quantum Jensen-Shannon Divergence Graph Kernels

In the previous Chapter we introduced a novel generative model for graphs. It is known, however, that discriminative classification approaches usually yield a higher prediction accuracy than generative ones, and thus in this Chapter we turn our attention to them. In particular, we introduce a novel kernel on unattributed and attributed graphs where we probe the graph structure through the evolution of a continuous-time quantum walk [55, 79]. Section 5.1 provides a brief overview of continuous-time quantum walks and other quantum-mechanical tools that will be used in the definition of our kernel. Section 5.4 introduces a novel graph kernel where we take advantage of the fact that the interference effects which characterise the quantum walk evolution seem to be enhanced by the presence of symmetrical motifs in the graph [51]. To this end, we define a walk onto a new structure that is maximally symmetric when the original graphs are isomorphic. To define the kernel, we make use of the quantum Jensen-Shannon divergence, a measure which has recently been introduced as a means to compute the distance between quantum states [94, 86]. Finally, in Section 5.5 we propose to apply standard manifold learning techniques on the kernel embedding to map the data onto a low-dimensional space where the different classes can exhibit a better linear separation.

## 5.1 Quantum Mechanical Background

The continuous-time quantum walk [55] is a natural quantum analogue of the classical random walk. Classical random walks model a diffusion process on a graph, and have proven to be a useful tool in the analysis of its structure. Let $G(V,E)$ be an undirected graph, where $V$ is a set of $n$ vertices and $E = (V \times V)$ is a set of edges. Diffusion on the graph is modeled as a Markovian process defined over $V$, with transitions restricted to adjacent vertices. More formally, we define the general state for the walk at time $t$ as a probability distribution over $V$, i.e., a vector, $\vec{p}_t \in \mathbb{R}^n$, whose $u$th entry gives the probability that the walk is at vertex $u$ at time $t$. Recall that the adjacency matrix of the graph $G$ is the symmetric matrix with elements

$$A_{uv} = \begin{cases} 1 \text{ if } (u,v) \in E \\ 0 \text{ otherwise} \end{cases} \tag{5.1}$$

and let $D$ be the diagonal matrix with elements $d_u = \sum_{v=1}^{n} A(u,v)$, where $d_u$ is the degree of the node $u$. Then, the *continuous-time random walk* on $G$ will evolve according to the equation

$$\vec{p}_t = e^{-Lt}\vec{p}_0 \tag{5.2}$$

where $L = D - A$ is the graph Laplacian, a combinatorial analogue of the Laplace-Beltrami operator [75].

The *continuous-time quantum walk*, i.e., the quantum counterpart of the continuous-time random walk, is similarly defined as a dynamical process over the vertices of the graph. By contrast to the classical case where the state vector is constrained to lie in a probability space, here the state of the system is defined through a vector of complex amplitudes over $V$ whose squared norm sums to unity over the nodes of the graph, with no restriction on their sign or complex phase. These phase differences allow interference effects to take place. Moreover, in the quantum case the evolution of the state vector of the walker is governed by a complex valued unitary matrix, whereas the dynamics of the classical random walk is governed by a stochastic matrix. Hence the evolution of the quantum walk is reversible, implying that quantum walks are non-ergodic and do not possess a limiting distribution. As a result, the behaviour of classical and quantum walks differs significantly, and quantum walks possess a number of interesting properties not exhibited by classical random walks.

More formally, using the Dirac notation, we denote the basis state corresponding to the walk being at vertex $u \in V$ as $|u\rangle$. A general state of the walk is a complex linear combination of the basis states, such that the state of the walk at time $t$ is defined as

$$|\psi_t\rangle = \sum_{u \in V} \alpha_u(t)\,|u\rangle \tag{5.3}$$

where the amplitude $\alpha_u(t) \in \mathbb{C}$ and $|\psi_t\rangle \in \mathbb{C}^{|V|}$ are both complex.

At each instant in time the probability of the walker being at a particular vertex of the graph is given by the square of the norm of the amplitude of the relative state. Let $X^t$ be a random variable giving the location of the walker at time $t$. Then the probability of the walker being at the vertex $u$ at time $t$ is given by

$$\Pr(X^t = u) = \alpha_u(t)\alpha_u^*(t) \tag{5.4}$$

where $\alpha_u^*(t)$ is the complex conjugate of $\alpha_u(t)$. Moreover $\sum_{u \in V} \alpha_u(t)\alpha_u^*(t) = 1$ and $\alpha_u(t)\alpha_u^*(t) \in [0,1]$, for all $u \in V$, $t \in \mathbb{R}^+$.

The evolution of the walk is then given by the Schrödinger equation, where we take the time-independent Hamiltonian of the system to be the graph Laplacian, yielding

$$\frac{\partial}{\partial t}|\psi_t\rangle = -iL|\psi_t\rangle\,. \tag{5.5}$$

Given an initial state $\left|\psi_0\right\rangle$, we can solve Equation (5.5) to determine the state vector at time $t$

$$\left|\psi_t\right\rangle = e^{-iLt}\left|\psi_0\right\rangle . \tag{5.6}$$

Note that generally one may use any Hermitian operator as the Hamiltonian. Common choices are the graph adjacency matrix, the normalized Laplacian and the signless Laplacian.

Finally, we can compute the spectral decomposition of the graph Laplacian $L = \Phi\Lambda\Phi^\top$, where $\Phi$ is the $n \times n$ matrix $\Phi = (\phi_1|\phi_2|...|\phi_j|...|\phi_n)$ with the ordered eigenvectors $\phi_j$s of $L$ as columns and $\Lambda = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_j, ..., \lambda_n)$ is the $n \times n$ diagonal matrix with the ordered eigenvalues $\lambda_j$ of $L$ as elements, such that $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. Using the spectral decomposition of the graph Laplacian and the fact that $\exp[-iLt] = \Phi\exp[-i\Lambda t]\Phi^\top$ we can then write

$$\left|\psi_t\right\rangle = \Phi e^{-i\Lambda t}\Phi^\top\left|\psi_0\right\rangle . \tag{5.7}$$

### 5.1.1 Density Operator

The observation process for a quantum system is defined in terms of projections onto orthogonal subspaces associated with operators on the quantum state space called *observables*. Let $O$ be an observable of the system, with spectral decomposition

$$O = \sum_i a_i P_i \tag{5.8}$$

where the $a_i$ are the (distinct) eigenvalues of $O$ and the $P_i$ the orthogonal projectors onto the corresponding eigenspaces. An observation of a quantum state $\left|\psi\right\rangle$ is one of the eigenvalues $a_i$ of $O$, which is observed with probability

$$P(a_i) = \left\langle\psi\right| P_i \left|\psi\right\rangle \tag{5.9}$$

leaving the system in the state

$$\left|\bar{\psi}\right\rangle = \frac{P_i\left|\psi\right\rangle}{||P_i\left|\psi\right\rangle||}, \tag{5.10}$$

where $||\left|\psi\right\rangle|| = \sqrt{\left\langle\psi\middle|\psi\right\rangle}$ is the norm of the vector $\left|\psi\right\rangle$.

The *density operator* (or *density matrix*) is introduced in quantum mechanics to describe a system whose state is an ensemble of pure quantum states $\left|\psi_i\right\rangle$, each with probability $p_i$. The density operator of such a system is defined as

$$\rho = \sum_i p_i \left|\psi_i\right\rangle\left\langle\psi_i\right| . \tag{5.11}$$

Density operators are positive unit-trace matrices directly linked with the observables of the (mixed) quantum system. The expectation value of the measurement can be calculated from the density matrix $\rho$:

$$\langle O\rangle = \text{tr}\left(\rho O\right), \tag{5.12}$$

where tr is the trace operator. Similarly, the observation probability of $a_i$ can be expressed in terms of the density matrix $\rho$ as

$$P(a_i) = \mathrm{tr}(\rho P_i) \tag{5.13}$$

Finally, after the measurement, the corresponding density operator will be

$$\rho' = \sum_i P_i \rho P_i \tag{5.14}$$

### 5.1.2 Quantum Jensen-Shannon Divergence

In this Chapter we intend to use continuous-time quantum walks to probe the structure of graphs. In particular, we will compare suitably defined quantum walks in order to establish the degree of similarity between two graphs. To this end, for each walk we would like to study how the probability distribution over the state space varies with time. Unfortunately, when a measurement is made the wave function collapses and, with a probability equal to the squared norm of its amplitude, only one of the possible basis states is observed. In other words, if the state $|u\rangle$ is observed, after the measurement the new state of the quantum walk will be $|\psi\rangle = |u\rangle$. This implies that all further information previously contained in the state is lost and further measurements will not yield any additional information about the pre-measurement state. Hence we need to design an experiment that will allow us to analyze the behaviour of the quantum walk without causing the wave function collapse. In this Section we will review the quantum Jensen-Shannon divergence (QJSD) [93, 94, 86], a recently introduced distinguishability measure between quantum states.

The *von Neumann entropy* [105] $H_N$ of a mixture is defined in terms of the trace and logarithm of the density operator $\rho$

$$H_N = -\mathrm{tr}(\rho \log \rho) = -\sum_i \xi_i \ln \xi_i \tag{5.15}$$

where $\xi_1, \ldots, \xi_n$ are the eigenvalues of $\rho$. If $\langle \psi_i | \rho | \psi_i \rangle = 1$, i.e., the quantum system is a pure state $|\psi_i\rangle$ with probability $p_i = 1$, then the Von Neumann entropy $H_N(\rho) = -\mathrm{tr}(\rho \log \rho)$ is zero. On other hand, for a mixed state described by the density operator $\sigma$ we have a non zero Von Neumann entropy associated with it.

With the Von Neumann entropy to hand, the quantum Jensen-Shannon divergence between two density operators $\rho$ and $\sigma$ is defined as

$$D_{JS}(\rho, \sigma) = H_N\Big(\frac{\rho+\sigma}{2}\Big) - \frac{1}{2} H_N(\rho) - \frac{1}{2} H_N(\sigma) \tag{5.16}$$

This quantity is always well defined, symmetric and positive definite.

It can also be shown that $D_{JS}(\rho, \sigma)$ is bounded, i.e., $0 \leq D_{JS}(\rho, \sigma) \leq 1$. Let $\rho = \sum_i p_i \rho_i$ be a mixture of quantum states $\rho_i$, with $p_i \in \mathbb{R}^+$ such that $\sum_i p_i = 1$, then one can prove that

$$H_N(\sum_i p_i \rho_i) \leq H_S(p_i) + \sum_i p_i H_N(\rho_i) \tag{5.17}$$

where $H_S$ indicates the Shannon entropy and the equality is attained if and only if the states $\rho_i$ have support on orthogonal subspaces. By setting $p_1 = p_2 = 0.5$, we see that

$$D_{JS}(\rho,\sigma) = H_N\left(\frac{\rho+\sigma}{2}\right) - \frac{1}{2}H_N(\rho) - \frac{1}{2}H_N(\sigma) \leq 1 \tag{5.18}$$

Hence $D_{JS}$ is always less than or equal to 1, and the equality is attained only if $\rho$ and $\sigma$ have support on orthogonal subspaces.

Our interest in the quantum Jensen-Shannon divergence lies in the fact that it verifies several interesting properties which are required for a good distinguishability measure between quantum states [94, 86]. The problem of discriminating between two quantum states $|\phi\rangle$ and $|\psi\rangle$ of a given physical system is of central importance in quantum computation and quantum information, and it is based on the definition of a suitable distance measure. Recall that a function

$$d = \mathbb{X} \times \mathbb{X} \longrightarrow \mathbb{R} \tag{5.19}$$

defined over a set $\mathbb{X}$ is a distance if, for every $x, y \in \mathbb{X}$,

$$d(x,y) \geq 0 \text{ with } d(x,y) = 0 \Longleftrightarrow x = y \tag{5.20}$$

and it is symmetric, i.e.,

$$d(x,y) = d(y,x) \tag{5.21}$$

Moreover, $d$ is said to be a metric for $\mathbb{X}$ if it satisfies the triangle inequality

$$d(x,y) + d(y,z) \geq d(x,z) \tag{5.22}$$

for every $x, y, z \in \mathbb{X}$.

In his seminal paper, Wootters [148] investigates the problem of distinguishability and defines the concept of statistical distance between pure quantum states. Here the distance between two different preparations $|\phi\rangle$ and $|\psi\rangle$ of the same physical system is computed by counting the number of distinguishable states between $|\phi\rangle$ and $|\psi\rangle$. The main result of Wootters' work is to show that this distance is equal to the angle in Hilbert space between $|\phi\rangle$ and $|\psi\rangle$. As a consequence, Wootter's distance is defined as

$$d_W(|\phi\rangle, |\psi\rangle) = \arccos(|\langle\phi|\psi\rangle|), \tag{5.23}$$

where $|\langle\phi|\psi\rangle|$ denotes the modulus of the inner product for $\phi$ and $\psi$. It can be proved that this distance satisfies the triangle inequality and is thus a metric.

Wootters' work is fundamentally based on the extension of a distance over the space of probability distributions to the Hilbert space of pure quantum states. Similarly, attempts to define a distance measure between pure and mixed quantum states are typically based on the generalization of divergence or distance measures commonly used in the space of probability distributions. This is the case of the relative entropy [89], which is a generalization of information theoretic Kullback-Leibler divergence. However, the relative entropy is neither a distance, as it is not symmetric, nor does it not satisfy the triangle inequality, and, most importantly, it is unbounded.

The square root of the QJSD, on the other hand, is bounded, it is a distance and, as proved by Lamberti et. al [86], it satisfies the triangle inequality. In particular, the authors give a formal proof for the case of pure states, while for the case of mixed states they support their claim with numerical evidence. Note that alternative metrics have been proposed in the literature, such as the Bures distance [38], which is defined as

$$B(\rho,\sigma) = \sqrt{2}\left[1 - \mathrm{tr}\left((\rho^{1/2}\sigma\rho^{1/2})^{1/2}\right)\right]^{1/2}. \tag{5.24}$$

The Bures distance and the QJSD require the same number of observations, since they both need the full density matrices to be computed. However, the QJSD turns out to be faster to compute than the Bures distance. In fact, the latter involves taking the square root of matrices, usually computed through matrix diagonalization which scales as $O(n^3)$, where $n$ is the number of vertices in the graph. On the other hand, to compute the QJSD only the eigenvalues of $\rho$, $\sigma$ and $\frac{\rho+\sigma}{2}$ are needed, which can be computed in $O(n^2)$.

## 5.2 Preliminaries

Given two graphs $G_1(V_1,E_1)$ and $G_2(V_2,E_2)$, we want to measure their similarity by comparing the evolution of two suitably defined continuous-time quantum walks on the graphs. Let $|\psi_t\rangle = \sum_{u\in V}\alpha_u(t)|u\rangle$ be a continuous-time quantum walk on $G(V,E)$ at time $t$ where the Hamiltonian is defined to be the graph adjacency matrix $A$. We let the two quantum walk evolve until a time $T$ and we define the average density operator $\rho_T$ over this time as

$$\rho_T = \frac{1}{T}\int_0^T |\psi_t\rangle\langle\psi_t|\,\mathrm{d}t \tag{5.25}$$

In other words, we defined a mixed system with equal probability of being in any of the pure states defined by the quantum walk evolution. Thus, we are now able to compute the divergence between two quantum walks on $G_1$ and $G_2$ as the quantum Jensen-Shannon divergence between their density operators. We now establish the complexity of computing the density matrix.

Recall that $|\psi_t\rangle = e^{-iAt}|\psi_0\rangle$. Note, however, that the following equations hold independently of the choice of the Hamiltonian. We start by rewriting Eq. 6.29 as

$$\rho_T = \frac{1}{T}\int_0^T e^{-iAt}|\psi_0\rangle\langle\psi_0|e^{iAt}\,\mathrm{d}t \tag{5.26}$$

Since $e^{-iAt} = \Phi e^{-i\Lambda t}\Phi^\top$, we can rewrite the previous equation in terms of the spectral decomposition of the adjacency matrix,

$$\rho_T = \frac{1}{T}\int_0^T \Phi e^{-i\Lambda t}\Phi^\top|\psi_0\rangle\langle\psi_0|\Phi e^{i\Lambda t}\Phi^\top\,\mathrm{d}t \tag{5.27}$$

The $(r,c)$ element of $\rho_T$ can be computed as

$$\rho_T(r,c) = \frac{1}{T}\int_0^T \left(\sum_k \sum_l \phi_{rk} e^{-i\lambda_k t}\phi_{lk}\alpha_l(0)\right)\left(\sum_m \sum_n \alpha_m(0)^\dagger \phi_{mn} e^{i\lambda_n t}\phi_{cn}\right) \mathrm{d}t \tag{5.28}$$

Let $\bar{\psi}_k = \sum_l \phi_{lk}\alpha_l(0)$ and $\bar{\psi}_n = \sum_m \phi_{mn}\alpha_n(0)^\dagger$, then

$$\rho_T(r,c) = \frac{1}{T}\int_0^T \left(\sum_k \phi_{rk} e^{-i\lambda_k t}\bar{\psi}_k \sum_n \phi_{cn} e^{i\lambda_n t}\bar{\psi}_n\right) \mathrm{d}t \tag{5.29}$$

which can be finally rewritten as

$$\rho_T(r,c) = \sum_k \sum_n \phi_{rk}\phi_{cn}\bar{\psi}_k\bar{\psi}_n \frac{1}{T}\int_0^T e^{i(\lambda_n - \lambda_k)t} \mathrm{d}t \tag{5.30}$$

If we let $T \to \infty$, Eq. 5.30 further simplifies to

$$\rho_T(r,c) = \sum_{\lambda_k \in \tilde{\Lambda}} \sum_m \sum_n \phi(\lambda_k)_{r,m}\phi(\lambda_k)_{c,n}\bar{\psi}_m\bar{\psi}_n \tag{5.31}$$

where $\tilde{\Lambda}$ is the set of unique eigenvalues of $A$ and $\phi(\lambda_k)$ is the matrix whose columns are the eigenvectors associated with $\lambda_k$. As a consequence, we see that the complexity of computing the density matrix is upper bounded by that of computing the eigendecomposition of $\mathcal{G}$, i.e. $O(|\mathcal{V}|^3)$.

Note that the time-average density operator can also be rewritten in terms of the projectors on the eigenspaces of the unitary operator inducing the walk, as we will show in Section 6.1.1.

## 5.3 A Graph Kernel From Continuous-Time Quantum Walks

Let the initial state of the a continuous-time quantum walk on the unattributed graph $G(V,E)$ be

$$|\psi_0\rangle = \sum_{u \in V} \alpha_u(0)\,|u\rangle \tag{5.32}$$

where $\alpha_u(0) = \frac{d_u}{C}$ and $d_u$ is the degree of vertex $u$. In our first attempt to define a graph kernel using the QJSD, we propose to compute the divergence between two graphs as the QJSD between their density operators, denoted $\rho_T$ and $\sigma_T$ respectively. Then, we define the continuous-time quantum walk kernel $k_{CTQW}(G_1,G_2)$ as

$$k_{CTQW}(G_1,G_2) = \exp(-\lambda D_{JS}(\rho_T,\sigma_T)) \tag{5.33}$$

where $\lambda$ is a decay factor which satisfies $0 < \lambda < 1$. Here $\lambda$ is used to ensure that the large values do not tend to dominant the kernel value. Note that this kernel is parametrized by the time $T$. For ease of computation, we decide to let $T \to \infty$.

**Lemma 5.3.1.** *The continuous-time quantum walk kernel is positive definite.*

*Proof.* **Proof** This follows the definitions in [86, 94, 83]. In [83], a diffusion kernel $k_s = \exp(\lambda s(G_p, G_q))$ associated with any symmetric similarity measure $s(G_p, G_q)$ has been proven to be positive definite. Since the quantum Jensen-Shannon divergence between a pair of density matrices is symmetric [86, 94], the proposed quantum Jensen-Shannon graph kernel is positive definite. □

When the graphs have different size, i.e., $|V_1| \neq |V_2|$, in order to compute $H_N\left(\frac{\rho_T + \sigma_T}{2}\right)$ we extend the smaller graph to the size of the larger one by adding a number of disconnected nodes. It is important to note that the resulting similarity measure is not permutation invariant. For this reason, in the next Section we will propose an alternative kernel which overcomes the problem by performing all the computation on a union of the two original graphs. First, however, we evaluate the classification performance of this kernel.

### 5.3.1 Experimental Evaluation

The experiments are performed on three different standard dataset, namely MUTAG, Enzymes and PPI. Table 5.2 reports some statistics about these datasets. MUTAG is a dataset of 188 mutagenic aromatic and heteroaromatic compounds labeled according to whether or not they have a mutagenic effect on the Gram-negative bacterium *Salmonella typhimurium*. Enzymes is a dataset of graphs representing protein tertiary structures that consists of 600 enzymes from the BRENDA enzyme database. Finally, the PPI dataset consists of protein-protein interaction (PPIs) networks related to histidine kinase from two different groups: 40 PPIs from *Acidovorax avenae* and 46 PPIs from *Acidobacteria*.

We then compare the performance of our kernel with several alternative methods, namely the Weisfeiler-Lehman subtree kernel [124], the shortest path graph kernel [31], the Shannon entropy associated with the information functionals FV and FP [49], and the Ihara zeta function on graphs [113]. For the kernel methods, we compute the kernel matrix of each graph kernel on each dataset and then we apply the kernel PCA [118] on the kernel matrix to embed the graphs into principle component space as feature vectors. For other methods, we compute the characteristics values of graphs on each dataset. We perform 10-fold cross-validation using a Support Vector Machine (SVM) with Sequential Minimal Optimization (SMO). All the SMO-SVMs and their parameters were optimized on a Weka workbench [147]. We report the average classification accuracy of each method in Table 5.1.

Despite the fact that our kernel is not permutation invariant, it is still competitive when compared with alternative methods. Note, however, that these results are achieved by first embedding the graphs onto a vectorial space using kPCA. Without this first step, the classification accuracy of our kernel actually turns out to be among

| Method | MUTAG | Enzymes | PPI |
|---|---|---|---|
| CTQW | 84.04 | 32.16 | **76.20** |
| SP | **85.29** | 31.16 | 72.92 |
| WL | 82.05 | **46.42** | 75.90 |
| FV | 84.57 | 24.17 | 70.93 |
| FP | 84.57 | 24.17 | 70.93 |
| ZF | 80.85 | 32.00 | 70.93 |

Table 5.1: Classification accuracy on unattributed graph datasets. CTQW is the proposed kernel, SP is the shortest-path kernel [31], WL is the Weisfeiler-Lehman subtree kernel [124], FV and FP are the information functionals [49] and ZF denotes the Ihara zeta function on graphs [113].

the worst. In the next Section, however, we show how to develop a permutation invariant QJSD kernel and we show that it outperforms the alternative kernels in a number of classification tasks.

## 5.4 QJSD Kernel

Given two graphs $G_1(V_1, E_1, v_1)$ and $G_2(V_2, E_2, v_2)$, where $v_1$ and $v_2$ are respectively the functions assigning attributes to the nodes of $G_1$ and $G_2$, we build a new graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ where $\mathcal{V} = V_1 \cup V_2$, $\mathcal{E} = E_1 \cup E_2 \cup E_{12}$, and $(u, v) \in E_{12}$ only if $u \in V_1$ and $v \in V_2$ (see Fig. 5.1 for an example). Moreover, the edges $(u, v) \in E_{12}$ are labeled with a real value $\omega(v_1(u), v_2(v))$ representing the similarity between $v_1(u)$ and $v_2(v)$. Note that in the case in which the graphs are unattributed, $\mathcal{G}$ will be unweighted. With this new structure to hand, we define two continuous-time quantum walks $|\psi_0^-\rangle = \sum_{u\in V} \alpha_u^-(0) |u\rangle$ and $|\psi_0^+\rangle = \sum_{u\in V} \alpha_u^+(0) |u\rangle$ on $\mathcal{G}$ with starting states

$$\alpha_u^-(0) = \begin{cases} +\frac{d_u}{C} \text{ if } u \in G_1 \\ -\frac{d_u}{C} \text{ if } u \in G_2 \end{cases} \qquad \alpha_u^+(0) = \begin{cases} +\frac{d_u}{C} \text{ if } u \in G_1 \\ +\frac{d_u}{C} \text{ if } u \in G_2 \end{cases} \tag{5.34}$$

where $d_u$ is the degree of the node $u$ and $C$ is the normalisation constant such that the probabilities sum to one.

Let the adjacency matrix of the graph be the Hamiltonian of the system. We then let the two quantum walks evolve until a time $T$ and we define the average density operators $\rho_T$ and $\sigma_T$ over this time as

$$\rho_T = \frac{1}{T}\int_0^T |\psi_t^-\rangle\langle\psi_t^-| \, \mathrm{d}t \qquad \sigma_T = \frac{1}{T}\int_0^T |\psi_t^+\rangle\langle\psi_t^+| \, \mathrm{d}t \tag{5.35}$$

Note that if the two original graphs are attributed, the walk on the composite structure will spread at a speed proportional to the edge weights, which means that given

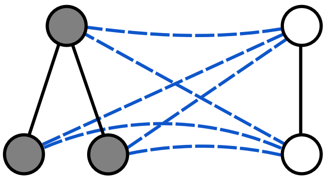


Figure 5.1: Given two graphs $G_1(V_1,E_1)$ and $G_2(V_2,E_2)$ we build a new graph $\mathscr{G}=(\mathscr{V},\mathscr{E})$ where $\mathscr{V}=V_1\cup V_2$, $\mathscr{E}=E_1\cup E_2$ and we add a new edge $(u,v)$ between each pair of nodes $u\in V_1$ and $v\in V_2$.

an edge $(u,v)\in E_{12}$, the more similar $\mathrm{v}_1(u)$ and $\mathrm{v}_2(v)$ are, the faster the walker will propagate along the inter graphs connection $(u,v)$. On the other hand, the intra-graph connection weights, which are not dependent on the nodes similarity, will not affect the propagation speed.

Then, given two unattributed graphs $G_1$ and $G_2$, we define the quantum Jensen-Shannon kernel $k_T(G_1,G_2)$ between them as

$$k_T(G_1,G_2)=D_{JS}(\rho_T,\sigma_T) \tag{5.36}$$

where $\rho_T$ and $\sigma_T$ are the density operators defined as in Eq. 6.29. Note that this kernel is parametrised by the time $T$. As it is not clear how we should set this parameter, here we propose to let $T\to\infty$. However, in Section 5.5.1 we will show that a proper choice of $T$ can yield an increased average accuracy in an SVM classification task.

We now proceed to show some interesting properties of our kernel. First, however, we need to prove the following lemma and theorem.

**Lemma 5.4.1.** *Given a graph G with adjacency matrix A, the unitary operator* $U^t=e^{-iAt}$ *is invariant to graph symmetries.*

*Proof.* Recall that $U^t=e^{-iAt}$, where $A$ is the graph adjacency matrix. If $u$ and $v$ belong to a symmetry orbit (a group of vertices where $v_1$ and $v_2$ belong to the same orbit if there is an automorphism $\tau\in\mathrm{Aut}(G)$ such that $\tau(v_1)=v_2$), then there exists an automorphism of the graph with a corresponding permutation matrix $\mathscr{P}$ such that

$$A=\mathscr{P}^\top A\mathscr{P} \tag{5.37}$$

and

$$\mathscr{P}|u\rangle=|v\rangle \tag{5.38}$$

In other words, the graph Laplacian is invariant to symmetries. As we will show, the same holds for the unitary operator of the quantum walk. In fact, given the spectral decomposition of the graph adjacency matrix $A=\Phi\Lambda\Phi^\top$, we can see that the following equality holds

$$\Phi\Lambda\Phi^\top=\mathscr{P}^\top(\Phi\Lambda\Phi^\top)\mathscr{P} \tag{5.39}$$

and thus

$$\Phi=\mathscr{P}^\top\Phi \tag{5.40}$$

Let us now write the unitary operator in terms of the adjacency matrix eigendecomposition, which yields

$$e^{-iAt} = \Phi e^{-i\Lambda t}\Phi^\top \tag{5.41}$$

From Equations 5.40 and 5.41 it follows that

$$\Phi e^{-i\Lambda t}\Phi^\top = \mathscr{P}^\top \Phi e^{-i\Lambda t}\Phi^\top \mathscr{P} \tag{5.42}$$

which concludes the proof. □

**Theorem 5.4.2.** *If $G_1$ and $G_2$ are two isomorphic graphs, then $\rho_T$ and $\sigma_T$ have support on orthogonal subspaces.*

*Proof.* We need to prove that

$$(\rho_T)^\dagger \sigma_T = \frac{1}{T^2}\int_0^T \rho_{t_1}\,\mathrm{d}t_1 \int_0^T \sigma_{t_2}\,\mathrm{d}t_2 = \mathbf{0} \tag{5.43}$$

where $\mathbf{0}$ is the matrix of all zeros, $\rho_t = |\psi_t^-\rangle\langle\psi_t^-|$ and $\sigma_t = |\psi_t^+\rangle\langle\psi_t^+|$. Note that if $\rho_{t_1}^\dagger \sigma_{t_2} = \vec{0}$ for every $t_1$ and $t_2$, then $\rho^\dagger\sigma = \mathbf{0}$. We now prove that if $G_1$ is isomorphic to $G_2$ then $\left\langle \psi_{t_1}^- \middle| \psi_{t_2}^+ \right\rangle = 0$ for every $t_1$ and $t_2$.

If $t_1 = t_2 = t$, then

$$\langle \psi_0^- |\, (U^t)^\dagger U^t \,| \psi_0^+ \rangle = 0 \tag{5.44}$$

since $(U^t)^\dagger U^t$ is the identity matrix and the initial states are orthogonal by construction. On the other hand, if $t_1 \neq t_2$, we have

$$\langle \psi_0^- |\, U^{\Delta t} \,| \psi_0^+ \rangle = 0 \tag{5.45}$$

where $\Delta_t = t_2 - t_1$.

To conclude the proof we rewrite the previous equation as

$$\begin{aligned}
\langle \psi_0^- |\, U^{\Delta t} \,| \psi_0^+ \rangle &= \sum_{k=1}^{n} \alpha_k^-(0) \sum_{l=1}^{n} \alpha_l^+(0) U_{lk}^{\Delta t} \\
&= \sum_{k_1=1}^{m} \alpha_{k_1}^+(0) \sum_{l_1=1}^{n} \alpha_{l_1}^+(0) U_{l_1 k_1}^{\Delta t} - \sum_{k_2=m+1}^{n} \alpha_{k_2}^+(0) \sum_{l_2=1}^{n} \alpha_{l_2}^+(0) U_{l_2 k_2}^{\Delta t} \\
&= \sum_{k_1=1}^{m} \sum_{l_1=1}^{n} \alpha_{k_1}^+(0) \alpha_{l_1}^+(0) U_{l_1 k_1}^{\Delta t} - \sum_{k_2=m+1}^{n} \sum_{l_2=1}^{n} \alpha_{k_2}^+(0) \alpha_{l_2}^+(0) U_{l_2 k_2}^{\Delta t} = 0
\end{aligned} \tag{5.46}$$

where the indices $l, l_1, l_2, k$ run over the nodes of $\mathscr{G}$, while $k_1$ and $k_2$ run over the nodes $G_1$ and $G_2$ respectively.

To see that Eq. 6.9 holds, note that according to Lemma 5.4.1 $U$ is invariant to graph symmetries, and that if $G_1$ and $G_2$ are isomorphic, the first and the second halves of $|\psi_0^+\rangle$ are equal up to the permutation which maps the nodes of $G_1$ to those of $G_2$. □

**Corollary 5.4.3.** *Given a pair of graphs $G_1$ and $G_2$, the kernel satisfies the following properties: 1) $0 \leq k_T(G_1, G_2) \leq 1$ and 2) if $G_1$ and $G_2$ are isomorphic, then $k_T(G_1, G_2) = 1$.*

*Proof.* The first property is trivially proved by noting that, according to Eq. 5.36, the kernel between $G_1$ and $G_2$ is defined as the quantum Jensen-Shannon divergence between two density operators, and then recalling that the value of quantum Jensen-Shannon divergence is bounded to lie between 0 and 1.

The second property follows again from Eq. 5.36 and Theorem 5.4.2. It is sufficient to note that the quantum Jensen-Shannon divergence reaches its maximum value if and only if the density operators have support on orthogonal spaces. □

Unfortunately we cannot prove that our kernel is positive semidefinite, but both empirical evidence and the fact that the Jensen-Shannon Divergence is negative semidefinite on pure quantum states [32] while our graph kernel is maximal on orthogonal states suggest that it might be.

### 5.4.1 Experiments on Unattributed Graphs

In this Section, we evaluate the performance of our kernel and we compare it with a number of well-known alternative graph kernels, namely the classic random walk kernel [60], the shortest-path kernel [31] and a set of graphlet kernels [125]. We test different variants of the graphlet kernel, where we vary the graphlet sizes $k \in \{3, 4\}$ and the type of graphlets (all possible size $k$ graphlets vs only those which are fully connected).

The experiments are performed on three different standard dataset, namely MUTAG, Enzymes and PPI. To these three datasets, we add a fourth set of 30 synthetically generated graphs, 10 for each class. The graphs belonging to each class were sampled from a generative model with size 12,14 and 16 respectively. Details about the generative model can be found in Chapter 4.

We first evaluate the Multidimensional Scaling embedding of the synthetic graphs for three different distance matrices, namely the edit distance, the distance between the graph spectra and the distance corresponding to our kernel function. The distance between the graph spectra is computed as follows. For each graph $G$ with adjacency matrix $A$, we compute the column vector $s_G$ of the ordered eigenvalues of $A$. As the graphs are of different sizes and thus their spectra are of different lengths, the vectors are all made to be the same length by padding zeros to the end of the shorter vector.

| datasets | # graphs | # classes | avg # nodes | disjoint |
|---|---:|---|---:|---:|
| Synth | 30 | 3 (10 each) | 13.77 | N |
| MUTAG | 188 | 2 (125 vs. 63) | 17.93 | N |
| Enzymes | 600 | 6 (100 each) | 32.63 | Y |
| PPI | 86 | 2 (40 vs. 46) | 109.60 | N |

Table 5.2: Statistics on the graph datasets.

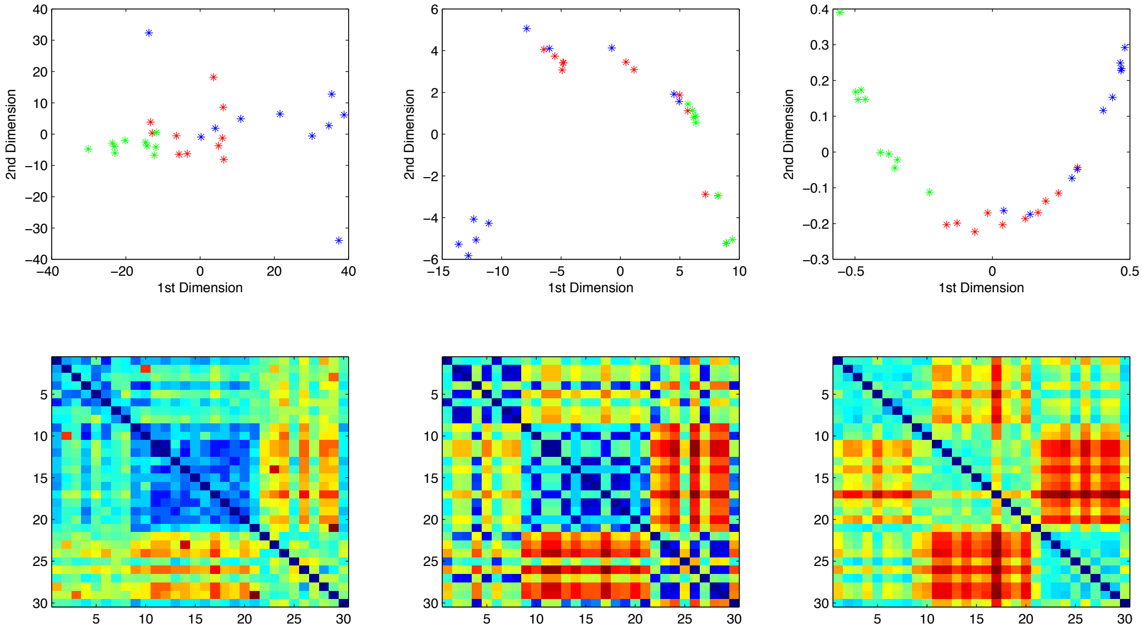


Figure 5.2: Two-dimensional MDS embeddings of the synthetic data (top row) on different distance matrices (bottom row). From left to right, the distance is computed as the edit distance between the graphs, the distance between the graph spectra and the distance associated with the QJSD kernel.

The $(i, j)$th element of the distance matrix is then $d_{ij} = ||s_i - s_i||$. Figure 5.2 shows the MDS embeddings and the graph distance matrices. It is clear that the distance matrix associated with our kernel has a well-defined block structure which is reflected in the MDS embedding, where the three classes seem to be easily separable.

A second experiment uses a binary C-SVM to test the efficacy of our kernel for classification. We perform 10-fold cross validation, where for each sample we independently tune the value of C, the SVM regularizer constant, by considering the training data from that sample. The process is averaged over 100 random partitions of the data. Given this setting, we first investigate the effect of the time parameter in the classification accuracy. Fig. 5.3 shows the value of the average accuracy (± standard error) on the synthetic dataset as the time parameter $T$ varies. Here the red horizontal line shows the mean accuracy for $T \to \infty$. The plot shows that the choice of the time greatly influences the performance of our kernel, as we can clearly see that the average accuracy reaches a maximum before stabilizing around the asymptotic value. This should be compared with the average accuracy that we achieve for $T \to \infty$, which, although not optimal, is not too far from the maximum.

Finally, Table 5.4 reports the average classification accuracies (± standard error) of the different kernels. As we can see, the proposed kernel achieves the best result on three out of four datasets. The poor accuracy on the Enzymes dataset is likely to be linked to the presence of disjoint graphs, as this will affect the way in which the walk

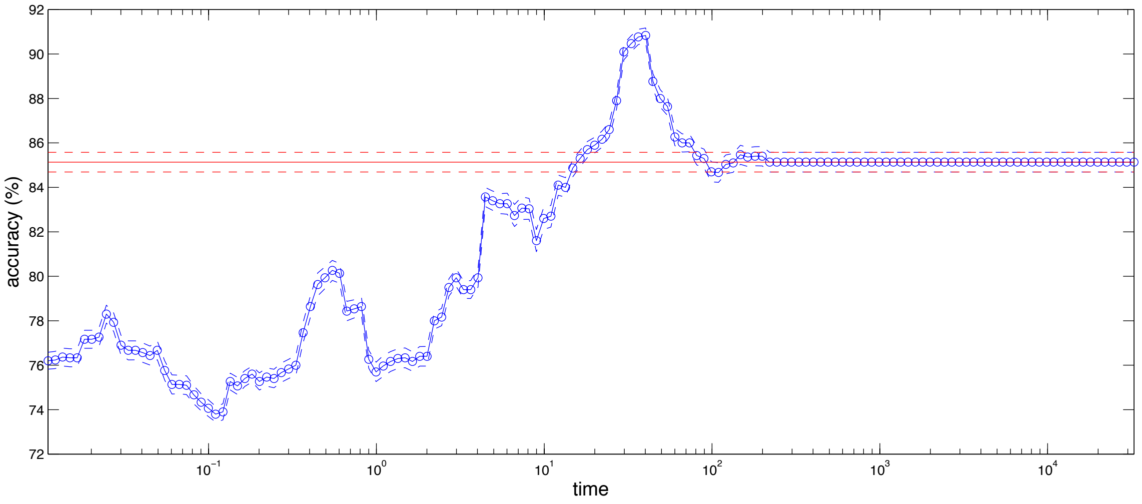


Figure 5.3: The mean accuracy (± standard error) of the QJSD kernel as the time parameter $T$ varies. The red horizontal line shows the mean accuracy for $T \to \infty$.

spreads through the graph. Note, however, that this is a particularly hard dataset where the structures of the graphs provide limited information about the underlying class structure. In fact, all kernels based only on graph structure perform only marginally better than random guess, and node and edge attributes need to be taken into account too.

### 5.4.2 Experiments on Attributed Graphs

In this Section, we evaluate the performance of the proposed kernel and we compare it with a number of well-known alternative graph kernels, namely the classic random walk kernel [60], the shortest-path kernel [31] and the 3-nodes graphlet kernel [125], both in their unattributed and attributed versions. Note that since the attributed versions of these kernels are defined only on graphs with categorically labeled nodes, in our experiments we will need to bin the node attributes before computing the kernels.

We use a binary C-SVM to test the efficacy of the kernels. We perform 10-fold cross validation, where for each sample we independently tune the value of C, the SVM regularizer constant, by considering the training data from that sample. The process is averaged over 100 random partitions of the data, and the results are reported in terms of average accuracy ± standard error.

#### Synthetic Data

We start by evaluating the proposed kernel on a set of synthetically generated graphs. To this end, we have randomly generated 3 different weighted graph prototypes with size 16, 18 and 20 respectively. For each prototype we started with an empty graph and then we iteratively added the required number of nodes each labeled with a random

| Kernel | Synth | MUTAG | Enzymes | PPI |
|---|---|---|---|---|
| QJSD | **85.20 ± 0.47** | **86.55 ± 0.15** | 24.20 ± 0.38 | **78.43 ± 0.30** |
| SP | 74.90 ± 0.33 | 85.02 ± 0.17 | **28.55 ± 0.42** | 66.14 ± 0.40 |
| RW | 78.53 ± 0.43 | 77.87 ± 0.21 | 22.15 ± 0.37 | 69.70 ± 0.30 |
| $G_3$ | 79.33 ± 0.39 | 82.04 ± 0.14 | 24.87 ± 0.22 | 51.95 ± 0.44 |
| $G_4$ | 83.60 ± 0.48 | 81.89 ± 0.13 | 28.60 ± 0.21 | 73.14 ± 0.37 |
| $CG_3$ | 56.57 ± 0.47 | 66.43 ± 0.08 | 19.92 ± 0.27 | 52.89 ± 0.50 |
| $CG_4$ | 81.57 ± 0.54 | 69.08 ± 0.15 | 23.05 ± 0.06 | 61.56 ± 0.41 |

Table 5.3: Classification accuracy (± standard error) on unattributed graph datasets. QJSD is the proposed kernel, SP is the shortest-path kernel [31], RW is the random walk kernel [60], while $G_k$ ($CG_k$) denotes the graphlet kernel computed using all graphlets (all the connected graphlets, respectively) of size $k$ [125].

mean and variance. Then we added the edges and their associated observation probabilities up to a given edge density. Given the prototypes, we sampled 20 observations from each class being careful to discard graphs that were disconnected. Details about the generative model used to sample the graphs can be found in Chapter 4. Figure 5.4 shows the edit distance matrix of the dataset and the Multidimensional Scaling of the graph distances.

With the synthetic graphs to hand, we initially investigate how the value of the kernel between two graphs varies as we apply Erdös-Rényi noise to the graph structure. In this case the similarity between two nodes $u$ and $v$ is defined as $\omega(u,v) = e^{-\lambda(v_1(u)-v_2(v))^2}$, where $v_1(u)$ and $v_2(v)$ are the real-valued attributes associated with $u$ and $v$ respectively. Figure 5.5 shows the result of this experiment. Here we randomly

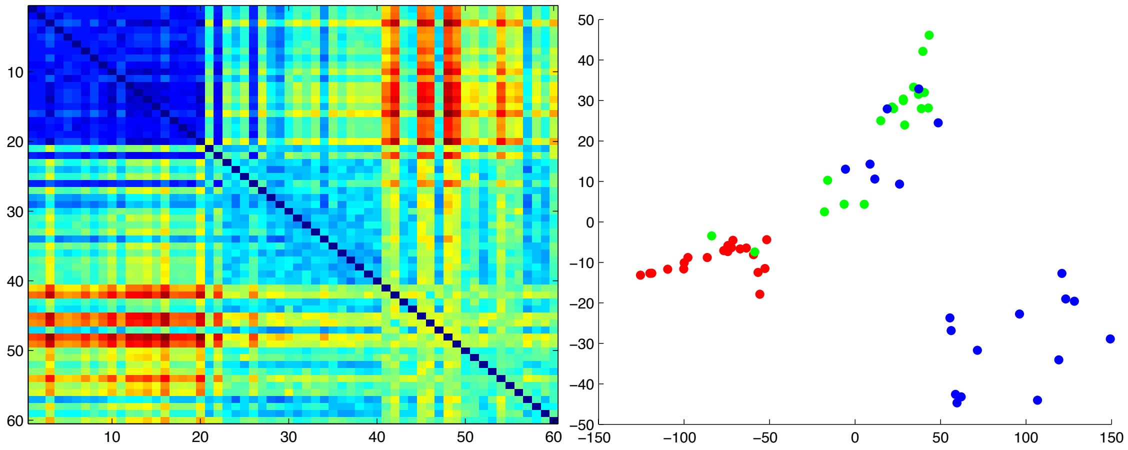
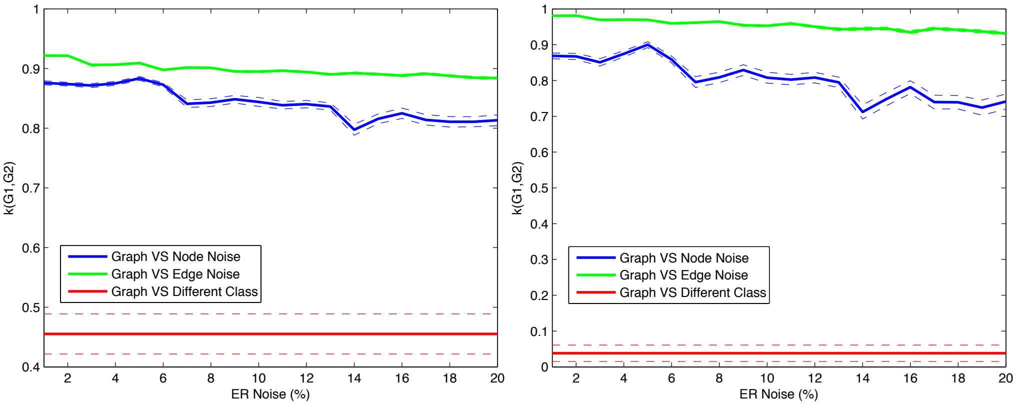


Figure 5.4: Edit distance matrix and Multidimensional Scaling of the graph distances for the synthetic dataset.



(a) Without Attributes

(b) With Attributes

Figure 5.5: The effects of Erdös-Rényi structural noise applied to the nodes and edges of the graph on the kernel value. Using the proposed similarity measure, the noisy versions of the graph belonging to the first class are clearly distinguishable from the instances of the second class. As expected, taking the attributes into account (right) makes the distinction even clearer (note the difference in the scale).

pick a graph $G$ belonging to class 1, and we compute a number of increasingly noisy versions of it. The noise is applied either to the edges only, i.e. adding or deleting edges, or to the nodes as well, i.e. adding or deleting nodes and edges. We then compute the average value of the kernel between $G$ and its corrupted versions, and we plot it against the average similarity between $G$ and the graphs of class 2. Figure 5.5 shows that, even at considerably high levels of noise, $G$ is clearly distinguishable from the instances of the second class. As expected, taking the attributes into account renders the distinction even clearer (note the change in the y-scale). However, when augmented with the attributes information, our kernel measure seems to be slightly more sensitive to structural noise, in particular when the noise is affecting the nodes of the graph.

As a second experiment, we test the accuracy of our kernel in a classification task. The results are shown in Table 5.4. As we can see, our kernel outperforms or is competitive with the alternatives, and yields a close to 100% average accuracy. Note also that, as expected, taking the similarity between the node attributes into account results in a marked increase in the kernel performance. Quite surprisingly, however, we found that the random walk kernel on the categorically labeled graphs yields a lower performance than its unattributed version.

### Delaunay Graphs

We then tested the efficacy of the proposed kernel on the COIL [100] dataset, which consists of images of different objects, with 72 views of each object obtained from equally spaced viewing directions over 360°. For each image, a graph is obtained by computing the Delaunay triangulation of the corner points extracted by the Harris

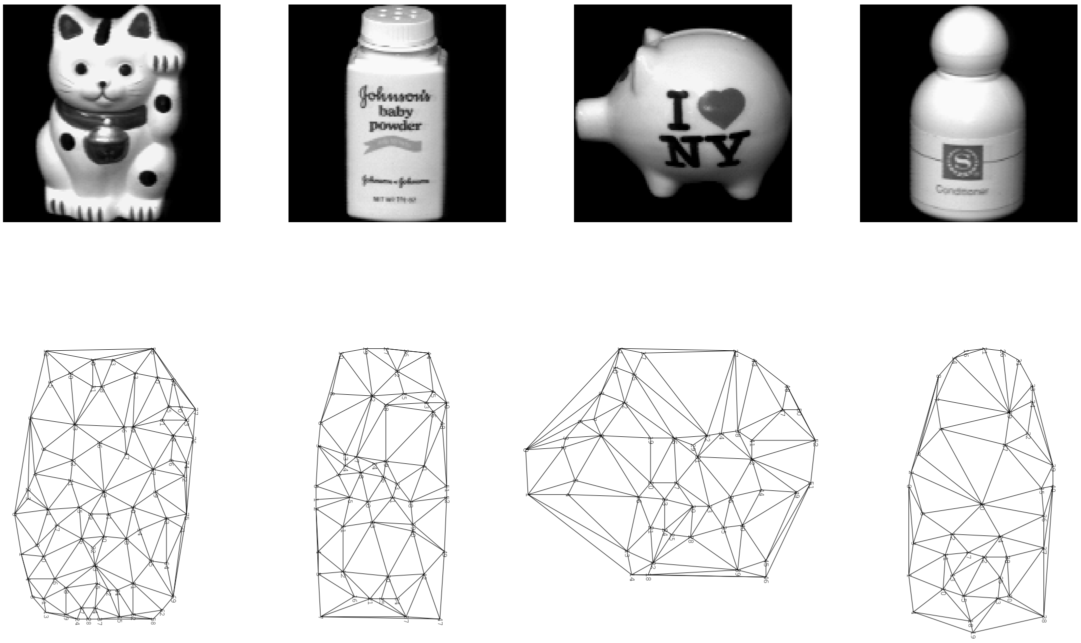


Figure 5.6: The four selected objects from the COIL [100] dataset and a sample of their associated Delaunay graphs. Each node of the graphs is labeled with the $(x, y)$ coordinates of the corresponding feature point.

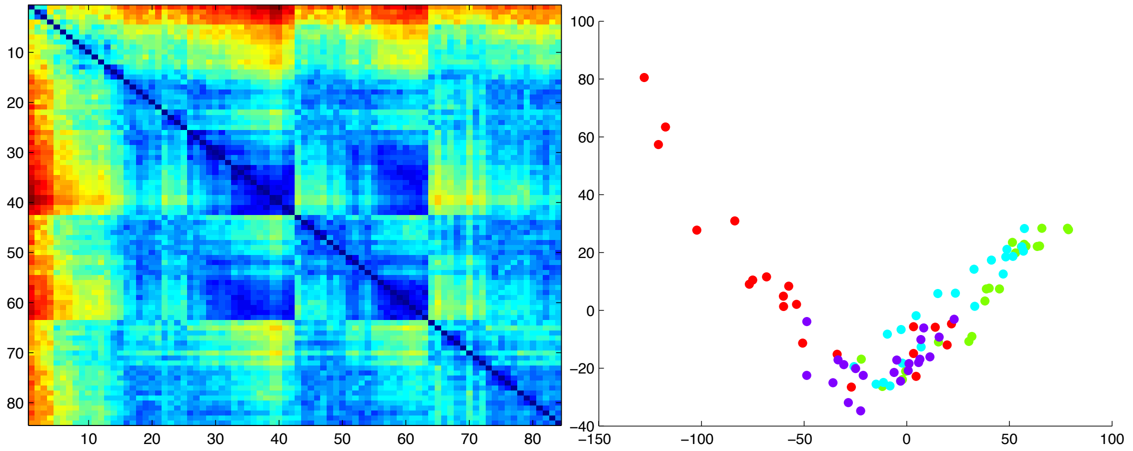


Figure 5.7: Edit distance matrix and Multidimensional Scaling of the graph distances for the COIL dataset.

corner detection algorithm. Moreover, each node is labeled with the $(x, y)$ coordinates of the corresponding feature point. The similarity between two nodes is $\omega(u, v) = e^{-\lambda||v_1(u)-v_2(v)||_2^2}$, where $||v_1(u) - v_(v)||_2$ is the Euclidean distance between the two feature points $u$ and $v$. Here we choose 4 different objects, each with 21 different 5° rotated views. Figure 5.6 shows the four selected objects together with their associated graphs, while Figure 5.7 shows the edit distance matrix and the MDS of the graph distances.

We first investigate how integrating the information on the nodes attributes influences the expressive power of our kernel. Figure 5.8 shows the MDS embedding on the graph distances computed from the unattributed kernel (left) and the attributed one (right). Although the embedding shows that a considerable overlap remains between the different classes, taking the node attributes similarities into account adds a further dimension which can help to discriminate better among the 4 selected objects.

This is indeed reflected in the results of the classification task shown in Table 5.4. In the attributed case, in fact, the average accuracy of the QJSD kernel is increased by more than 10%, and it outperforms that of all the remaining kernels. Note, however, that if the node labels are dropped, the performance of the QJSD kernel is among the lowest, which once again underlines the importance of incorporating the attributes similarities in the compositional structure.

### Shock Graphs

Finally, we experimented using shock graphs, a skeletal-based representation of the differential structure of the boundary of a 2D shape. We extracted graphs from a database composed of 120 shapes divided into 8 classes of 15 shapes each. Each graph has a node attribute that reflects the size of the boundary feature generating the corresponding skeletal segment. Figure 5.9 shows the shape database, the edit distances matrix between the shock graphs and the corresponding MDS. As we can see, the class structure is not very clear, and there is a considerable overlap between different classes. This is reflected in the average accuracy of the kernels, which is the lowest among the three datasets, as Table 5.4 shows. However, the proposed kernel still outperforms or is competitive with the others.

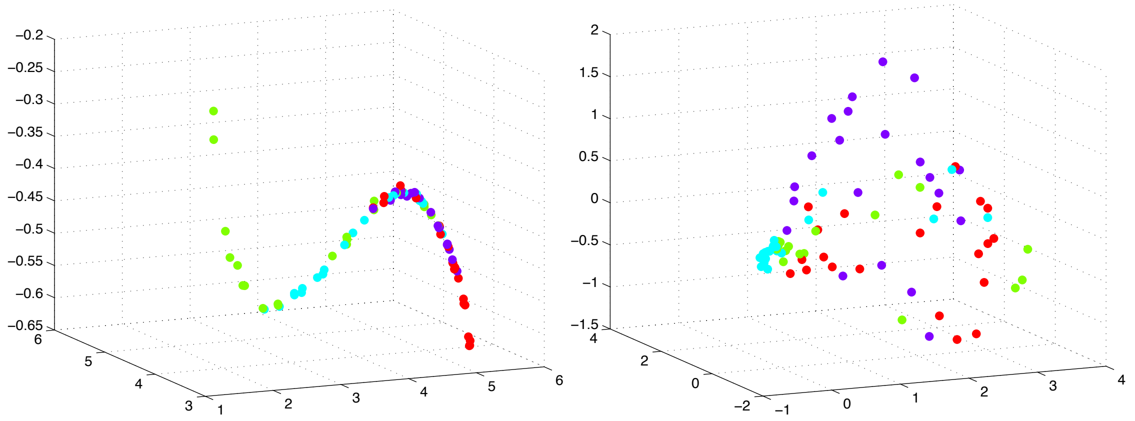


Figure 5.8: Multidimensional Scaling of the graph distances computed from the kernel matrix of the COIL dataset. Left, completely structural approach; right, including the information on the nodes attributes.

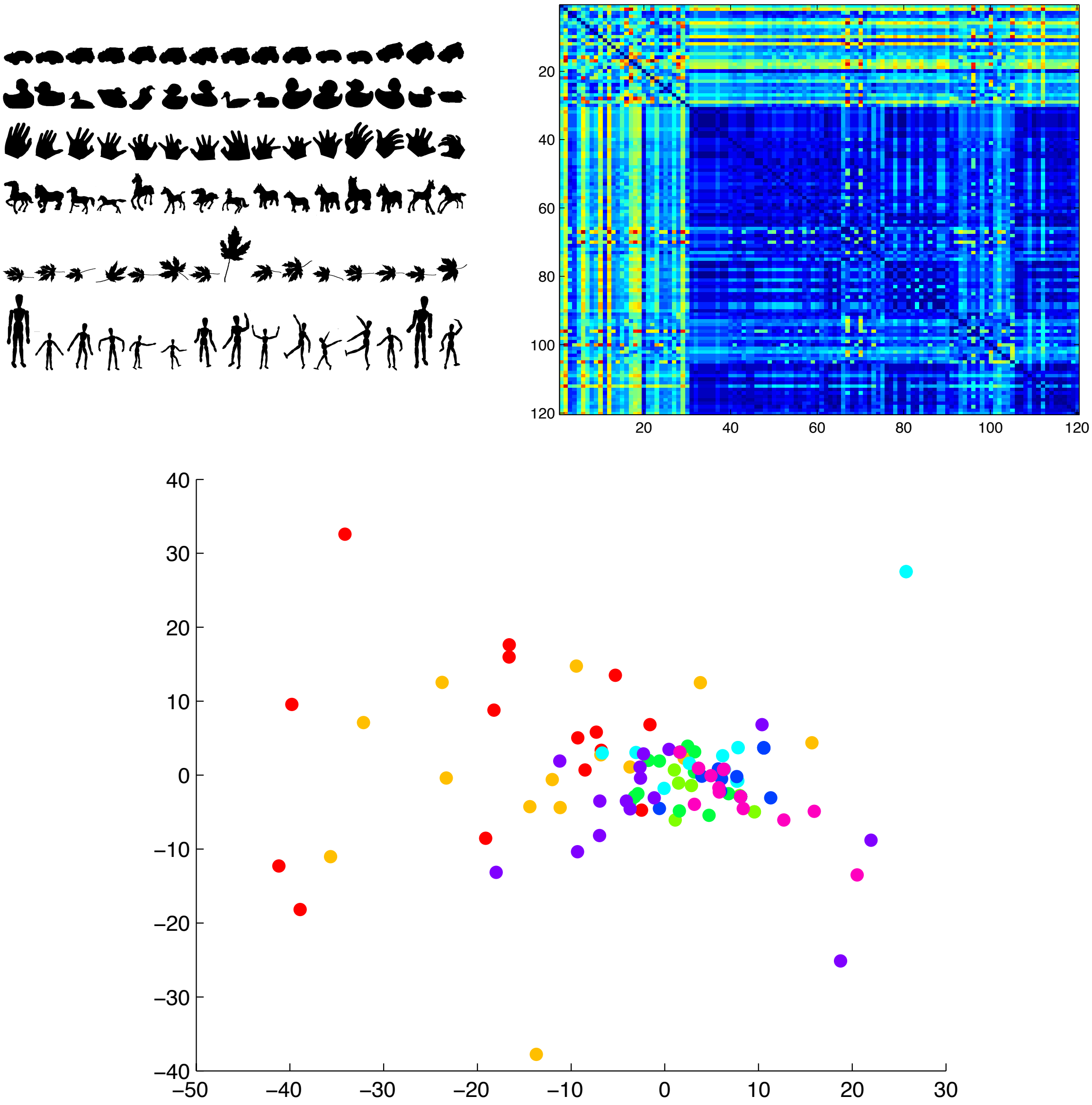


Figure 5.9: Top row: Left, a sample of the shape database; right, edit distance matrix. Bottom row: Multidimensional Scaling of the edit distances. As we can see, the class structure is not very clear and there is a considerable overlap between different classes.

## 5.5 Manifold Learning on the QJSD Kernel

In this Section, we study the separability properties of the QJSD kernel and we apply standard manifold learning techniques [133, 48] on the kernel embedding to map the data onto a low-dimensional space where the different classes can exhibit a better linear separation. The idea stems from the observation that the multidimensional scaling

| Kernel | Synth | Shock | COIL |
|---|---|---|---|
| QJSD$_w$ | 95.87 ± 0.14 | **66.65 ± 0.22** | **95.56 ± 0.20** |
| QJSD | 84.57 ± 0.25 | 53.97 ± 0.19 | 84.05 ± 0.22 |
| SP$_w$ | **96.36 ± 0.12** | 65.05 ± 0.25 | 94.40 ± 0.14 |
| SP | 91.13 ± 0.15 | 52.62 ± 0.32 | 85.25 ± 0.21 |
| RW$_w$ | 92.97 ± 0.18 | 53.26 ± 0.29 | 90.78 ± 0.26 |
| RW | 80.23 ± 0.30 | 26.11 ± 0.32 | 78.60 ± 0.25 |
| G3$_w$ | 88.75 ± 0.25 | 41.18 ± 0.27 | 89.25 ± 0.21 |
| G3 | 85.60 ± 0.25 | 38.85 ± 0.32 | 84.20 ± 0.22 |

Table 5.4: Classification accuracy (± standard error) on attributed graph datasets. QJSD is the proposed kernel, SP is the shortest-path kernel of Borgwardt and Kriegel [31], RW is the random walk kernel of Gartner et al. [60], while $G_3$ denotes the graphlet kernel computed using all graphlets of size 3 described in Shervashidze et al. [125]. The subscript $w$ identifies the kernels which make use of the attributes information. The best performing kernel for each dataset is highlighted in bold.

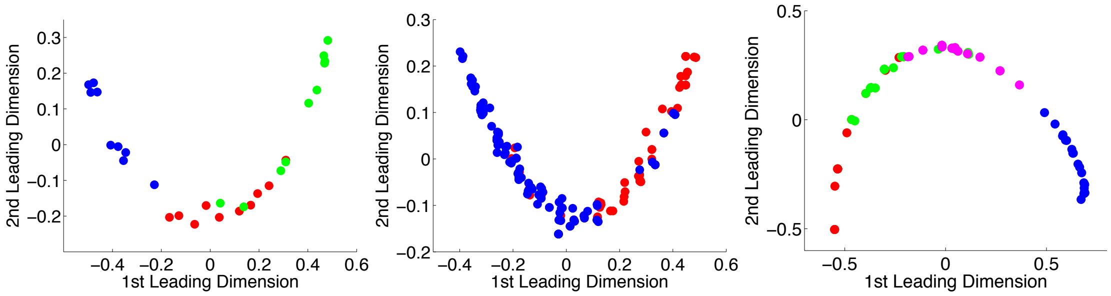


Figure 5.10: The MDS embeddings from the QJSD kernel consistently show an horse-shoe shape distribution of the points.

embeddings of the QJSD kernel show the so-called *horseshoe effect* [80]. This particular behaviour is known to arise when long range distances are not estimated accurately, and it implies that the data lie on a non-linear manifold. This is no surprise, since Emms et al [52] have shown that the continuous-time quantum walk underestimates the commute time related to the classical random walk. For this reason, it is natural to investigate the impact of the locality of distance information on the performance of the QJSD kernel. Given a set of graphs, we propose to use Isomap [133] to embed the graphs onto a low-dimensional vectorial space, and we compute the separability of the graph classes as the distance information varies from local to global. Moreover, we perform the same analysis on a set of alternative graph kernels commonly found in the literature [60, 31, 125]. Experiments on several standard datasets demonstrate that the Isomap embedding shows a higher separability of the classes.

Figure 5.10 shows the MDS embedding of the distance matrices associated with the

QJSD kernel for the synthetic, MUTAG and COIL datasets. Details on the datasets can be found in Section 5.5.1. These embeddings clearly suffer from a horseshoe shape effect, which is usually the result of an accurate estimate of the distance between objects only when they are close together, but not when they are far apart [80]. As a consequence, it should be possible to increase the kernel performance by filtering out in some way this long range distance information.

Here we propose a simple yet effective way to achieve this goal. Given a set of graphs, we compute the Isomap [133] embedding of the graphs and we evaluate the separability of the graph classes as the distance information varies from local to global. Isomap is a well-known manifold learning technique, which extends classical MDS by incorporating the pairwise geodesic distances between points. To this end, a neighborhood graph is constructed from the original set of points, where each node is connected to its $k$ nearest neighbors in the high-dimensional space. The geodesic distance between two nodes is then defined as the sum of the edge weights along the shortest-path between them. It is known that Isomap suffers from several shortcomings, so further work should focus on experimenting with more robust manifold learning techniques.

The class separability is evaluated in the following way. For each embedding, we perform a 10-fold cross validation using a binary C-SVM with a linear kernel, where we let the value of the SVM regularizer constant C vary over the interval $10^{-3}$ and $10^{3}$. Then, we take the maximum value of the average classification accuracy as an indicator of the separability. More formally, we look for the Isomap embedding which maximizes

$$\arg\max_{d,k}\max_{C}\alpha \tag{5.47}$$

where $\alpha$ is the 10-fold cross validation accuracy of the C-SVM, $C$ is the regularizer constant, $d$ is the embedding dimension and $k$ is the number of nearest neighbors. Note that the multi-classification task is solved using majority voting on a set of one-vs-one C-SVM classifiers.

### 5.5.1 Experimental Results

The experiments are performed on four different dataset, namely MUTAG, PPI, COIL [100] and a set of shock graphs. The COIL dataset consists of the 4 objects shown in Figure 5.11, each with 72 views obtained from equally spaced viewing directions over 360°. For each image, a graph is obtained as the Delaunay triangulation of the Harris corner

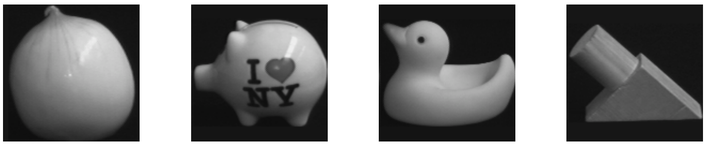
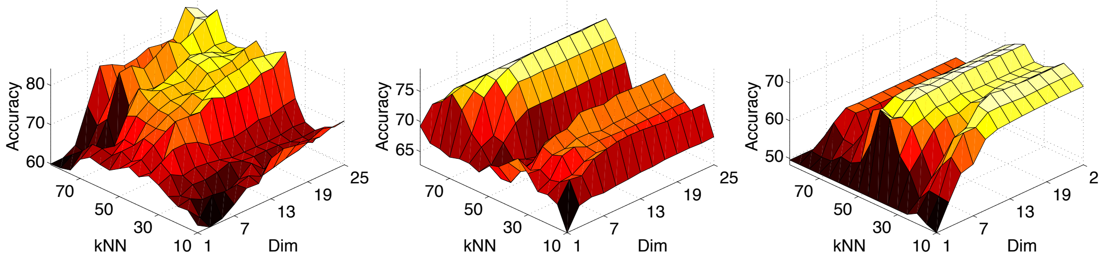


Figure 5.11: Sample images of the four selected object from the COIL-100 [100] dataset.



(a) QJSD (b) Random Walk (c) Graphlet

Figure 5.12: 3D plot of the 10-fold cross validation accuracy on the PPI dataset as the number of the nearest neighbors $k$ and the embedding dimension $d$ vary.

points. Finally, we select a set of shock graphs, a skeletal-based representation of the differential structure of the boundary of a 2D shape. The 120 graphs are divided into 8 classes of 15 shapes each. Each graph has a node attribute that reflects the size of the boundary feature generating the corresponding skeletal segment. To reflect the presence of attributes, the QJSD kernel is modified by labeling the new connections of the merged graph with the similarity between its two endpoints. To these four datasets, we add a fifth set of 30 synthetically generated graphs, 10 for each class. The graphs belonging to each class were sampled from a generative model with size 12,14 and 16.

Figure 5.12 shows the 3D plots of the 10-fold cross validation accuracy on the Isomap embeddings of the QJSD, the random walk and the graphlet kernels for the PPI dataset, as the size of the initial neighborhood and the embedding dimension vary. The plots show that for this dataset the QJSD kernel seems to be less sensitive to the locality of the distance information. On the other hand, for the graphlet kernel the maximum accuracy is achieved for a smaller neighborhood, which means that in this case the long range distance information is less accurate.

Figure 5.13 shows the two-dimensional Isomap embeddings with the highest linear separability for the QJSD kernels on the synthetic dataset, MUTAG and COIL. The result clearly shows the lack of the horseshoe shape distribution of Figure 5.10. Note, however, that the best embedding is usually found at a dimension higher than two

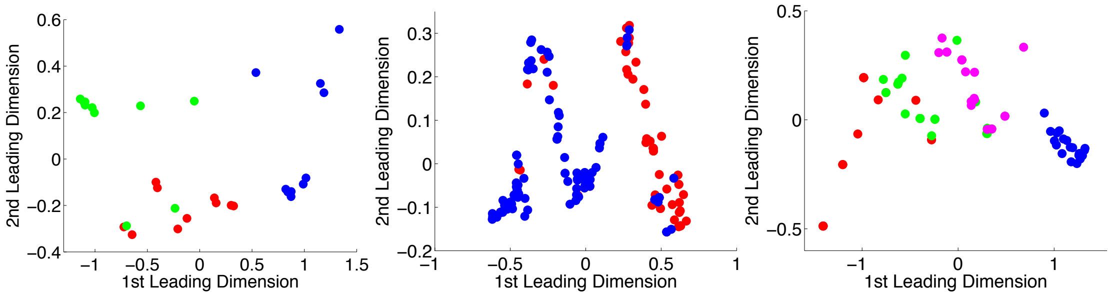


Figure 5.13: The optimal two-dimensional Isomap embeddings in terms of separability between the graph classes.

| Kernel | Synthetic | MUTAG | PPI | COIL | Shock |
|---|---|---|---|---|---|
| QJSD | *90.00* | *88.27* | *78.75* | 84.44 | *67.50* |
| QJSD$_{ISO}$ | **96.67** | **91.96** | **90.69** | **91.53** | **77.50** |
| SP | 80.00 | 86.08 | 71.25 | 85.56 | 61.67 |
| SP$_{ISO}$ | 86.67 | 89.33 | 87.08 | 89.17 | 60.05 |
| RW | 86.67 | 77.02 | 70.97 | 79.72 | 49.17 |
| RW$_{ISO}$ | 86.67 | 81.35 | 82.50 | 80.97 | 50.12 |
| GR | 86.67 | 82.92 | 49.56 | *86.67* | 39.17 |
| GR$_{ISO}$ | 90.00 | 84.53 | 77.08 | 87.78 | 54.17 |

Table 5.5: Maximum classification accuracy on the unattributed graph datasets. Here SP is the shortest-path kernel of Borgwardt and Kriegel [31], RW is the random walk kernel of Gartner et al. [60], while GR denotes the graphlet kernel computed using all graphlets of size 3 described in Shervashidze et al. [125], while the subscript $ISO$ indicates the result after the Isomap embedding. For each dataset, the best performing kernel before and after the embedding is shown in bold and italic, respectively.

and, as shown in Figure 5.12, the separability can change significantly as the dimension varies. Figure 5.13 also shows a clearer separation among the different classes, as highlighted in Table 5.5, which shows the separability of the data for each kernel and dataset. It is interesting to observe that, with the exception of a few cases, the Isomap embedding always yields an increased separability of the data, independently of the original kernel. It should also be underlined that the QJSD kernel always yields the highest separation, with a maximum classification accuracy above 90% in 4 out of 5 datasets.

## 5.6 Conclusions

In this Chapter, we have introduced a novel kernel on unattributed and attributed graphs where we probe the graph structure using the time evolution of a continuous-time quantum walk. More precisely, given a pair of graphs we computed the quantum Jensen-Shannon divergence between the evolution of two quantum walks on a suitably defined union of the original graphs. With the quantum Jensen-Shannon divergence to hand, we established our graph kernel. We performed an extensive experimental evaluation and we demonstrated the effectiveness of the proposed approach.

We then studied the separability properties of the QJSD kernel and we have proposed a way to compute a low-dimensional embedding where the separation of the different classes is enhanced. The idea stems from the observation that the multidimensional scaling embeddings on this kernel show a strong horseshoe shape distribution, a pattern which is known to arise when long range distances are not estimated accurately. Here we proposed to use Isomap to embed the graphs using only local distance information onto a new vectorial space with a higher class separability. An extensive experimental evaluation has shown the effectiveness of the proposed approach.

In the next Chapter, we will use the quantum mechanical framework introduced here to develop a set of novel algorithms for analyzing the structure of graphs. More precisely, we will show how to use the connection between structural symmetries and the interference effects of quantum walks to establish the presence of approximate axial symmetries in the graph and to define a new node centrality measure.

# 6 Graph Structure Analysis

The quantum mechanical analysis discussed in the previous Chapter is based on the fundamental connection between the structure of a graph and the interference effects which arise during the evolution of quantum walks. More precisely, the proposed framework was based on the existence of an intimate connection between structural symmetries and destructive (constructive) interference. In this Chapter we intend to investigate further the relation between quantum walks and graph symmetries, and, in particular, we are interested in exploiting the interference effects of quantum walks to probe the structure of a graph. In a sense, then, the concepts introduced here fit into the more general field of complex network science.

The remainder of this Chapter is organized as follows: Section 6.1 provides what appears to be the first attempt in the literature to measure the amount of approximate symmetries possessed by a graph. Section 6.2, on the other hand, is devoted to the explicit detection of approximate symmetry axes, but using a semi-classical rather than a purely quantum approach. Section 6.3 concludes the Chapter with the introduction of a novel vertex centrality index which relates the phase of a vertex to its influence on the evolution of a suitably defined quantum walk.

## 6.1 Measuring the Degree of Symmetry of a Graph

In this Section we attempt to quantify the degree of (approximate) symmetries possessed by a graph by evaluating the quantum Jensen-Shannon divergence [86, 94] between the evolution of two quantum walks on the graph with suitably defined initial states.

### 6.1.1 Quantum Mechanical Setup

Given a pair of nodes $u \in V$ and $v \in V$ in an undirected graph $G(V,E)$, we define two independent quantum walks with starting states

$$\left|\psi_0^-\right\rangle = \frac{|u\rangle - |v\rangle}{\sqrt{2}} \qquad \left|\psi_0^+\right\rangle = \frac{|u\rangle + |v\rangle}{\sqrt{2}}, \tag{6.1}$$

where, and to recap our earlier definition, the basis state corresponding to the walk being at vertex $u \in V$ is denoted as $|u\rangle$. Intuitively, by setting the initial amplitude on the two nodes to be respectively in anti phase and in phase, we allow the walk to highlight the presence of destructive and constructive interference patterns on the graph. We then let the two quantum walks evolve under Equation 5.6 until a time $T$ and we define the average density operators $\rho_T$ and $\sigma_T$ over this time as

$$\rho_T = \frac{1}{T}\int_0^T |\psi_t^-\rangle\langle\psi_t^-|\, \mathrm{d}t \qquad \sigma_T = \frac{1}{T}\int_0^T |\psi_t^+\rangle\langle\psi_t^+|\, \mathrm{d}t \tag{6.2}$$

where we use the graph Laplacian as the Hamiltonian of the system. In other words, our system has equal probability of being in any of the pure states $|\psi_t^-\rangle$ ($|\psi_t^+\rangle$ respectively) defined by the quantum walk evolution.

Given this setting, we are now able to compute the quantum Jensen-Shannon divergence $D_{JS}(\rho_T, \sigma_T)$ between the two walks using Equation 5.16. Due to the interference effect, we expect the mixed states for the two walks to have maximum divergence when the two initial nodes are symmetrically located in the graph. This is a consequence of the way in which we have initialised the two walks. Specifically, we aim to use the destructive and constructive interference effect by setting the initial node amplitudes to be respectively in anti phase and in phase. On the other hand, when the two nodes are not symmetrically located then we expect the two resulting mixed states to be similar, thus yielding a low value of $D_{JS}(\rho_T, \sigma_T)$. In the following theorem we prove that when $u$ and $v$ are symmetrically placed, then $\rho_T$ and $\sigma_T$ have support on orthogonal subspaces, which implies $D_{JS}(\rho_T, \sigma_T) = 1$.

**Theorem 6.1.1.** *Let $\rho_T$ and $\sigma_T$ be defined as in Equation 6.29. If $u, v$ are symmetrically placed and $|\psi_0^-\rangle$ and $|\psi_0^+\rangle$ are defined as in Equation 6.1, then $D_{JS}(\rho_T, \sigma_T) = 1$.*

*Proof.* We start by noting that if $\rho_T$ and $\sigma_T$ have support on orthogonal subspaces then

$$(\rho_T)^\dagger \sigma_T = \frac{1}{T^2}\int_0^T \rho_{t_1}\, \mathrm{d}t_1 \int_0^T \sigma_{t_2}\, \mathrm{d}t_2 = \vec{0} \tag{6.3}$$

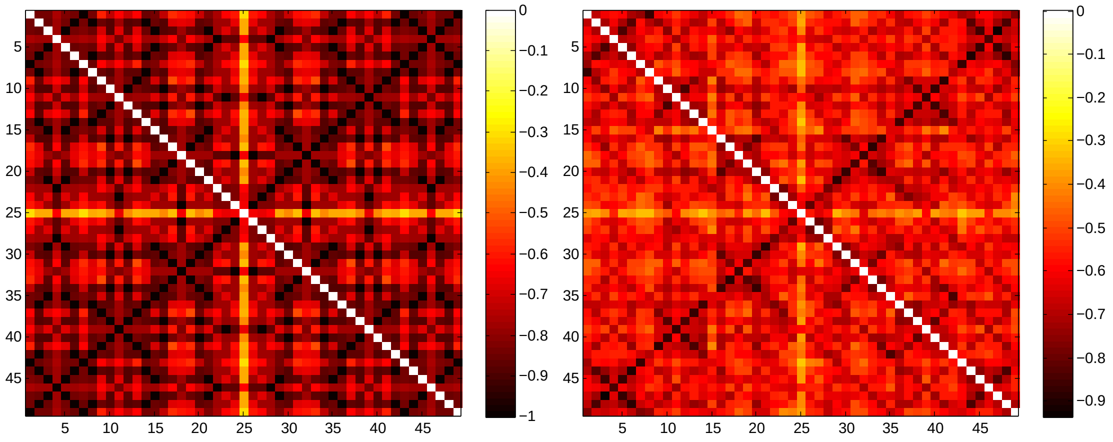
where $\vec{0}$ is the matrix of all zeros, $\rho_t = |\psi_t^-\rangle\langle\psi_t^-|$ and $\sigma_t = |\psi_t^+\rangle\langle\psi_t^+|$. Note that if $\rho_{t_1}^\dagger \sigma_{t_2} = \vec{0}$ for every $t_1$ and $t_2$, then $(\rho_T)^\dagger \sigma_T = \vec{0}$. We can hence go on to show that if $u$ and $v$ are symmetric, then $\left\langle \psi_{t_1}^- \middle| \psi_{t_2}^+ \right\rangle = 0$ for every $t_1$ and $t_2$. Let $U^t = e^{-iLt}$. If $t_1 = t_2 = t$, then

$$\langle\psi_0^-|\, (U^t)^\dagger U^t\, |\psi_0^+\rangle = 0 \tag{6.4}$$

since by definition $(U^t)^\dagger U^t$ is the identity matrix (since $U$ is unitary) and the initial states are orthogonal by construction.

On the other hand, if $t_1 \neq t_2$, we need to prove that when $u$ and $v$ are symmetrical then $|\psi_{t_1}^-\rangle$ and $\left|\psi_{t_2}^+\right\rangle$ are still orthogonal. In other words,

$$\langle\psi_0^-|\, U^{\Delta t}\, |\psi_0^+\rangle = 0 \tag{6.5}$$



(a) 7x7 Grid

(b) Noisy 7x7 Grid

Figure 6.1: The QJSD between pairs of walks initialised according to Equation 6.1. Here the color indicates the value of the QJSD between two walks and the axes are indexed by the nodes, where the 49 nodes of the grid are numbered from 1 to 49 from left to right, from top to bottom. Note that the QJSD of the two walks is maximum (equal to 1) when the two walks are initialized on symmetrically placed nodes. If the symmetry is broken by deleting one edge 6.1(b), the QJSD remains considerably higher on approximately symmetrically placed nodes.

where $\Delta_t = t_2 - t_1$. Recall that $\psi_0^- = 1/\sqrt{2}(|u\rangle - |v\rangle)$ and $\psi_0^+ = 1/\sqrt{2}(|u\rangle + |v\rangle)$. Then, if we denote by $U_{ij}^t$ the $ij$-th element of $U^t$, we have that

$$\langle \psi_0^- | U^{\Delta t} | \psi_0^+ \rangle = U_{uu}^{\Delta t} - U_{vv}^{\Delta t} + U_{uv}^{\Delta t} - U_{vu}^{\Delta t} \tag{6.6}$$

which further reduces to

$$\langle \psi_0^- | U^{\Delta t} | \psi_0^+ \rangle = U_{uu}^{\Delta t} - U_{vv}^{\Delta t} \tag{6.7}$$

since the matrix $U^t$ is symmetric.

To conclude the proof, we prove that when $u$ and $v$ are symmetrical we have $U_{uu}^t = U_{vv}^t$. This is immediate if we observe that Lemma 5.4.1 still holds if we replace the adjacency matrix with the graph Laplacian.

□

We should stress, however, that the converse of Theorem 6.2.1 does not hold. Note, in fact, that if we were able to prove the converse then we could give a polynomial-time solution to the graph isomorphism problem.

The proof of Theorem 6.2.1 basically relies on the fact that whenever two nodes $u$ and $v$ are symmetrical, then $U_{uu}^t = U_{vv}^t$ for each time $t$, where $U_{xx}^t$ is the wave kernel signature of $x$ at time $t$. However, our analysis relies only on computing the divergence between two density operators, while directly observing the wave kernel signature would cause a collapse of the wave function. Note also that a similar analysis can

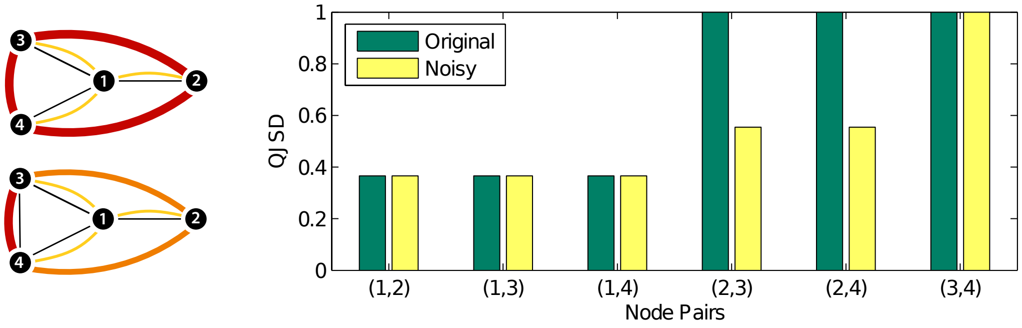


Figure 6.2: A star graph with 4 nodes and a modified version where two leaves are connected by an extra edge representing structural noise. The bar graph shows that although the symmetry between nodes 2-3 and nodes 2-4 is broken with the addition of an extra edge, the QJSD is still sensibly higher for those pairs of nodes, suggesting the presence of an approximate symmetry.

be done by comparing the heat kernel signature [132] $\vec{h}(x) = (H_{xx}^{t_1}, H_{xx}^{t_2}, \cdots, H_{xx}^{t_k})$ of $u$ and $v$, where we denote by $H_{xx}^t$ the solution of the heat equation at point $x$ at time $t$. On a manifold, it can be shown that if $H_{uu}^t = H_{vv}^t$ for each $t$, then the two points have the same global geometry, which means they either are the same point or symmetrically placed, with respect to the intrinsic geometry. Note, however, that this only holds for points on a manifold.

Figure 6.1 shows the value of $D_{JS}(\rho_T, \sigma_T)$ for all the possible pairs of nodes with initial non-zero amplitude on a $7 \times 7$ grid with reflecting boundary conditions. In the remainder of the Chapter we will refer to this matrix as the QJSD matrix. As expected, the QJSD matrix clearly reveals the presence of several perfect symmetries, i.e., pair of nodes for which $D_{JS}(\rho_T, \sigma_T) = 1$. Note that if we randomly delete an edge the symmetries are very likely to be broken, as we observe in Figure 6.1(b). Although we don't observe any perfect symmetry, the value of $D_{JS}(\rho_T, \sigma_T)$ remains higher on some pairs which were previously identified as being symmetrical, suggesting a connection between approximate symmetries and high values of the quantum Jensen-Shannon divergence.

To further support this claim, in Figure 6.9 we show the value of the QJSD for a star graph with four nodes and a noisy version of it, where the noise is represented by an additional edge joining nodes #3 and #4. Clearly, in the original star graph the three leaves are all symmetric with respect to the root node. However, if we alter the structure of the graph by adding an edge between #3 and #4, this results in breaking the symmetries between #2 and #3 and between #2 and #4 and, as a consequence, the QJSD between these nodes decreases. Interestingly, however, the QJSD for these pairs remains higher than the QJSD between #1 and #2, which is exactly what we would expect given the original symmetry.

### Efficient computation of the QJSD

In this sub-Section we show how to compute the solution to Equation 6.29 analytically. Let $P_\lambda = \sum_{k=1}^{\mu(\lambda)} \phi_{\lambda,k}\phi_{\lambda,k}^\top$ be the projection operator on the subspace spanned by the $\mu(\lambda)$ eigenvectors $\phi_{\lambda,k}$ associated with the eigenvalue $\lambda$ of the graph Laplacian. The evolution operator of the quantum walk can be then expressed in terms of this set of projectors, i.e.,

$$U^t = \sum_\lambda e^{-i\lambda t} P_\lambda \tag{6.8}$$

Recall that $|\psi_t\rangle = U^t |\psi_0\rangle$. According to Equation 6.8, we can rewrite the density operator $\rho_t$ associated with the pure state $|\psi_t\rangle$ as

$$\rho_t = U^t \rho_0 (U^t)^\dagger = \sum_{\lambda_1\in\Lambda} \sum_{\lambda_2\in\Lambda} e^{-i(\lambda_1-\lambda_2)t} P_{\lambda_1} \rho_0 P_{\lambda_2}^\top \tag{6.9}$$

As a consequence, we can reformulate Equation 6.29 as

$$\rho_T = \frac{1}{T}\int_0^T \rho_t \, \mathrm{d}t = \sum_{\lambda_1\in\Lambda} \sum_{\lambda_2\in\Lambda} P_{\lambda_1} \rho_0 P_{\lambda_2}^\top \frac{1}{T}\int_0^T e^{-i(\lambda_1-\lambda_2)t} \, \mathrm{d}t \tag{6.10}$$

Solving the integral in Equation 6.10 finally yields

$$\rho_T = \sum_{\lambda_1\in\Lambda} \sum_{\lambda_2\in\Lambda} P_{\lambda_1} \rho_0 P_{\lambda_2}^\top \frac{i(1-e^{iT(\lambda_2-\lambda_1)})}{T(\lambda_2-\lambda_1)} \tag{6.11}$$

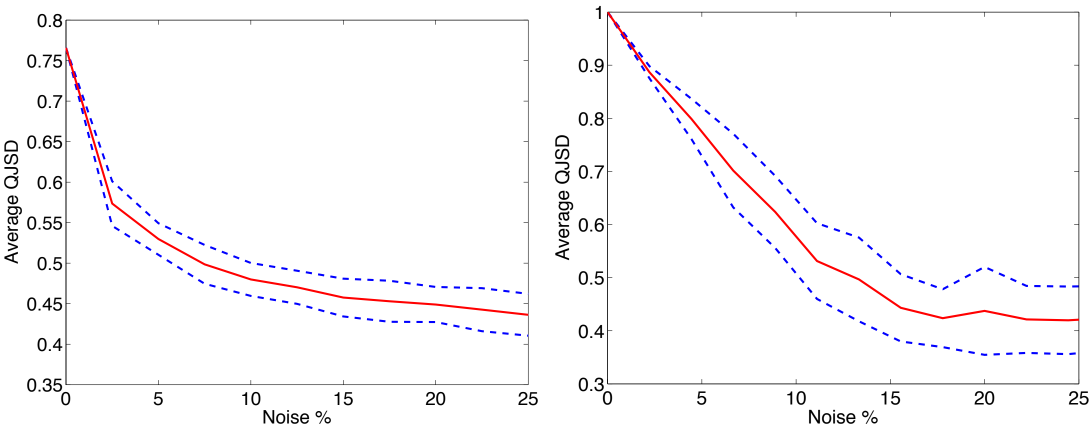
Note that if we let $T \to \infty$, then the integral in Equation 6.10 reduces to the Dirac delta function $\delta(\lambda_1 - \lambda_2)$. Hence, Equation 6.10 simplifies to

$$\rho_\infty = \sum_{\lambda\in\tilde{\Lambda}} P_\lambda \rho_0 P_\lambda^\top \tag{6.12}$$

where $\tilde{\Lambda}$ is the set of distinct eigenvalues of the graph Laplacian, i.e. the eigenvalues $\lambda$ with multiplicity $\mu(\lambda) = 1$. A consequence of Equation 6.12 is that the infinite-time limit of the average density matrix commutes with the graph Laplacian $L$, in fact

$$\begin{aligned} L\rho_\infty = \left(\sum_{\lambda\in\lambda} \lambda P_\lambda P_\lambda^\top\right)\left(\sum_{\lambda\in\lambda} P_\lambda \rho_0 P_\lambda^\top\right) = \sum_{\lambda\in\lambda} P_\lambda \lambda \rho_0 P_\lambda^\top = \\ = \left(\sum_{\lambda\in\lambda} P_\lambda \rho_0 P_\lambda^\top\right)\left(\sum_{\lambda\in\lambda} \lambda P_\lambda P_\lambda^\top\right) = \rho_\infty L. \end{aligned} \tag{6.13}$$

Hence, given the spectral decomposition of the graph Laplacian $L = \Phi\Lambda\Phi^\top$, the density matrix, expressed in the eigenvector basis given by $\Phi$, assumes a block diagonal form, where each block corresponds to an eigenspace of $L$ corresponding to a single eigenvalue. Thus, if $L$ has all eigenvalues distinct, then $\rho_\infty$ expressed in the unique eigenbasis of $L$ will be diagonal and its diagonal entries will directly correspond to



(a) 5 × 5 Grid

(b) Complete Graph

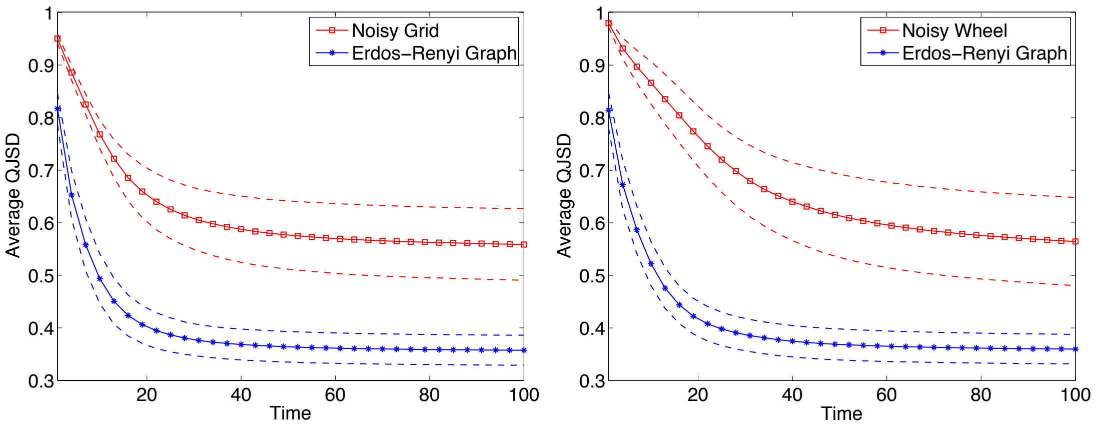
Figure 6.3: The average QJSD as a function of the structural (edge) noise for a 5 × 5 grid and a complete graph. Adding by randomly deleting (inserting) edges has the effect of breaking the symmetries of the original graphs and as a consequence the average QJSD decreases. Here the solid line indicates the mean, while the dashed lines indicate the standard deviation.

its eigenvalues. More generally, to compute the eigenvalues of $\rho_\infty$, we need to solve independently for the eigenvalues of each diagonal block, resulting in a complexity $O\left(\sum_{\lambda\in\tilde{\Lambda}}\mu(\lambda)^2\right)$, where $\mu(\lambda)$ is the multiplicity of the eigenvalue $\lambda$.

Note that Godsil [62] recently proved a similar result for the average mixing matrix. Here, the mixing matrix of a continuous-time quantum walk at time $t$ is defined as $M_t = (U^t)^\dagger \circ U^t$, where $U^t$ is the unitary operator inducing the walk and $(A \circ B)_{ij} = A_{ij}B_{ij}$ denotes the Schur-Hadamard product of two matrices $A$ and $B$. The average mixing matrix is then defined as the limit of the Cesaro mean of the mixing matrix as $t \to \infty$. Finally, the author shows that this quantity can be rewritten in terms of the projectors on the eigenspaces of the unitary operator of the walk.

### 6.1.2 Experimental Results

In this Section we intend to use the QJSD matrix to measure the degree of symmetry possessed by a graph. The basic requirements of this measure should be a) that its value increases (decreases) as the number of approximate symmetries of the graph increases (decreases), b) that it is permutation invariant and c) possibly easy to compute. Here we choose to use the average of the QJSD matrix as a simple yet effective means of characterizing the degree of symmetry possessed by a graph. Although it is known that as a statistic the average lacks robustness, since it is significantly affected by outliers, our experiments show that it provides a fast and permutation invariant way of measuring the degree of symmetry of a graph. More precisely, we investigate how the



(a) 5 × 5 Grid

(b) Wheel

Figure 6.4: The average of the QJSD matrix clearly distinguishes between a random graph and a symmetrical graph where artificial noise is added. Here the solid line indicates the mean, while the dashed lines indicate the standard deviation.

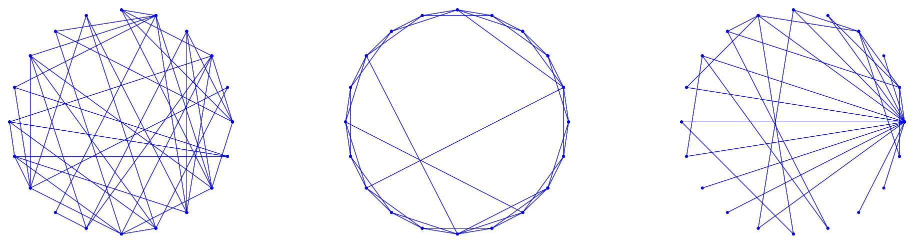
average QJSD over the pair of nodes varies for increasing time intervals. To this end, we numerically simulate the evolution of the two quantum walks with starting states as defined in Equation 6.1 using the software package MATLAB.

In our first experiment, we take a 5x5 grid with reflecting boundary conditions and a complete graph of size 10 and we iteratively add structural noise by deleting an increasing number of edges at each step. The procedure is repeated 100 times, and for each level of noise we compute the mean over the 100 trials of the average QJSD on the noisy graphs, where for each pair of nodes the QJSD is computed as in Equation 6.12. Figure 6.3 shows the result, where the structural noise affects from 0% to 25% of the graph edges. Here the solid line indicates the mean, while the dashed line indicates the standard deviation over the 100 repeated trials. Note that as the noise increases, the graphs become less and less symmetric, and at the same time the average QJSD rapidly decreases. This seems to fit with our hypothesis that the average QJSD can be used as a simple indicator of the degree of symmetry of a graph.

As a second experiment, we take the same 5x5 grid and we randomly create noisy versions of it by adding or deleting up to 3 edges at random locations. We then compare the average QJSD (over all pairs of nodes) on these graphs with that of a set of Erdös-Rényi random graphs. Figure 6.4 shows the average of the QJSD matrix for time intervals of increasing length. Again the solid line indicates the mean, while the dashed line indicates the standard deviation over 100 trials. As we can see, we are able to completely discriminate between the noisy versions of the 5x5 grid and the Erdös-Rényi graphs. This seems to confirm our intuition that the average QJSD matrix is able to capture the presence of (approximate) symmetrical patterns in a graph. We repeat the same experiment, but this time we perturb the 32-cycle graph where we have added a central axis of symmetry which connects an opposite pair of vertices. Again, the perturbed versions of the modified 32-cycle graph have a higher average QJSD when compared to Erdös-Rényi random graphs.

As a third experiment, we select three different random graphs models, namely the Watts-Strogatz [144], the Barabási-Albert [22] and the Erdös-Rényi [53] models. The Erdös-Rényi random graphs are generated by connecting pairs of nodes in the graphs with a uniform probability $p$. The Watts-Strogatz model produces small-world graphs with a high clustering coefficient and a short average path length. Finally, the preferential attachment algorithm of Barabási and Albert generates scale-free graphs. In this type of random graph the degree distribution of the vertices follows the power-law distribution, which is a property observed in many real-world graphs. In Figure 6.5, we show some examples of Erdös-Rényi, small-world and scale-free random graphs. We add to these three graph models a set of strongly regular graphs. A regular graph with $v$ vertices and degree $k$ is said to be strongly regular if there are two integers $\varepsilon$ and $\theta$ such that every two adjacent vertices have $\varepsilon$ common neighbours and every two non-adjacent vertices have $\theta$ common neighbors. We choose strongly regular graphs because they are known to be highly symmetric and this should be reflected in the value of the QJSD.

We can see from Figure 6.6 that we are able to discriminate these three types of random graphs by observing the average QJSD. In particular, due to their nature, the small-world graphs seem to have more symmetries than the two alternative models. In fact, the small-world graph is constructed by randomly linking the nodes of a regular ring lattice, thus yielding an interpolation between an Erdös-Rényi graph and a regular graph. Note also that the average QJSD is reduced by adding or deleting random edges, since this amounts to hiding the symmetrical patterns under increasing levels of noise. Although reduced, the average QJSD for the small-world graphs remains considerably higher than that of the Erdös-Rényi and scale-free graphs, where the addition of random noise does not seem to alter the average QJSD. As expected, the high number of symmetries possessed by strongly regular graphs is reflected in the higher value of the average QJSD, which remains clearly distinct from the three random graphs even in the presence of Erdös-Rényi noise. Note also that if the graph structure of the strongly

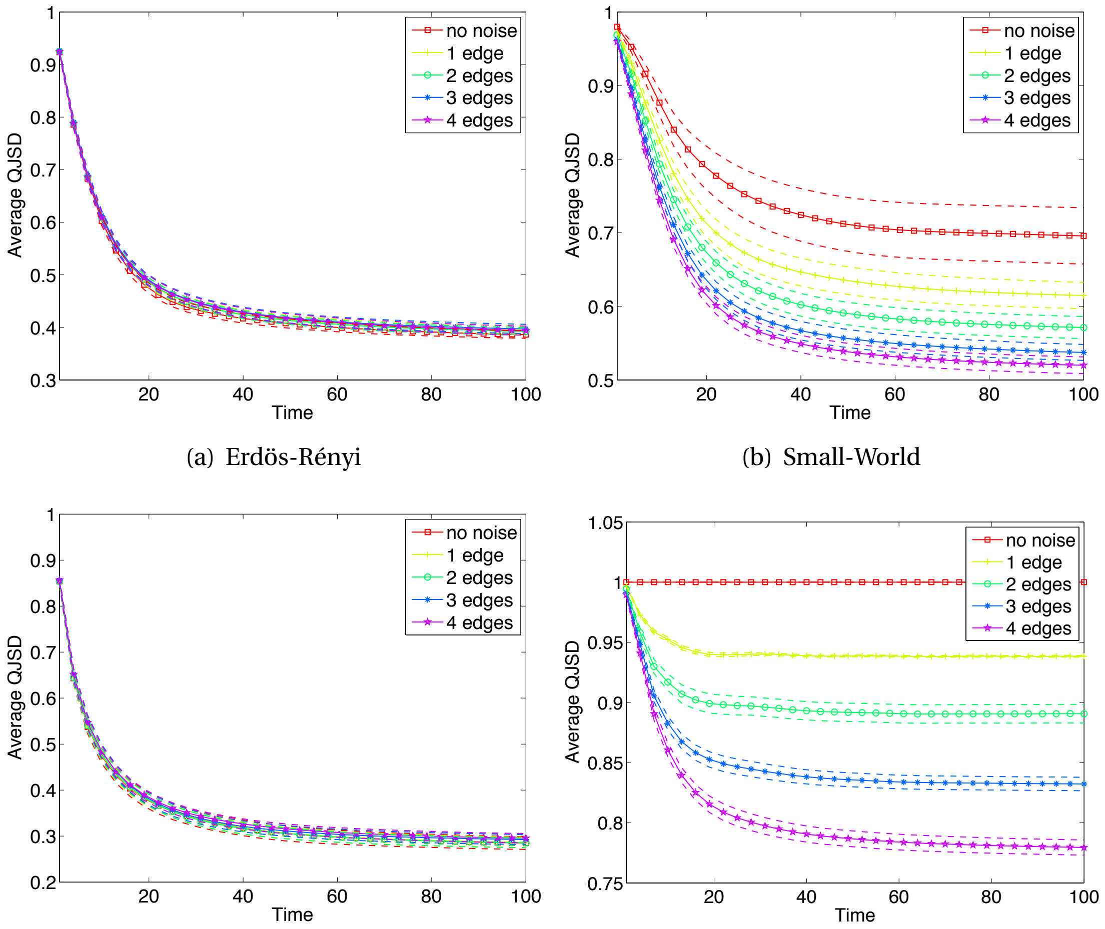


(a) Erdös-Rényi

(b) Small-World

(c) Scale-Free

Figure 6.5: Examples of graphs generated by the Erdös-Rényi, Watts-Strogatz and Barabási-Albert models respectively.



(a) Erdös-Rényi

(b) Small-World

(c) Scale-Free

(d) Strongly Regular

Figure 6.6: The effects of noise on the mean of the QJSD matrix on different type of graphs, for time intervals of increasing length. Note that here the solid line indicates the mean, while the dashed lines indicates the standard error.

regular graph is not perturbed, the QJSD between each pair of nodes is maximum, i.e. each pair of nodes is in a symmetrical relation. Finally, although the behaviour of the scale-free and Erdös-Rényi graphs is somewhat similar under noise, it is still possible to distinguish between them. In other words, the average QJSD of a scale-free graph is generally lower than that of an Erdös-Rényi graph.

## 6.2 Approximate Axial Symmetries Detection

We now turn to the problem of explicitly identifying the approximate axes of symmetry of a graph. To this end, however, we need to somehow relax the quantum mechanical formalism adopted so far, and turn to a more semi-classical approach. Although not clearly stated, in fact, the analysis described in the previous Section is indeed not semi-classical, as it is fully based on observable properties. This is because although the QJSD is not directly a quantum-mechanical observable, it can be computed from density matrices whose entries are indeed observables. Here, however, we require the complete observation of the wavefunction evolution in order to detect the nodes of the symmetry axes.

As a first attempt one may try to detect approximate axial symmetries by simply measuring the destructive interference patterns of continuous-time quantum walks. More precisely, given a suitable starting state for the quantum walk, one may identify those nodes of the graph which show a low average observation probability as the axial nodes. However, this analysis doesn't take into account the fact that a low observation probability may be also due to other structural characteristics, such as a low degree of the node.

A better solution requires evolving two quantum walks on the graph rather than just one. Similarly to the previous Section, these two walks are initialised so as to highlight the destructive and constructive interference patterns. With this setting to hand, those vertices that simultaneously show a low observation probability under destructive interference and a high observation probability under constructive interference are identified as axial.

### 6.2.1 Quantum Mechanical Setup

Given a pair of vertices $u, v$, we define again two independent quantum walks on $G$ with starting states

$$\left|\psi_0^-\right\rangle = \frac{|u\rangle - |v\rangle}{\sqrt{2}} \qquad \left|\psi_0^+\right\rangle = \frac{|u\rangle + |v\rangle}{\sqrt{2}} \tag{6.14}$$

Let us denote by $\alpha_v^-(t)$ and $\alpha_v^+(t)$ the amplitude on node $v$ at time $t$ during the evolution of the quantum walk with initial state $\left|\psi_0^-\right\rangle$ and $\left|\psi_0^+\right\rangle$, respectively. According to Eq. 6.14 we have that $\alpha_u^-(0) = -\alpha_v^-(0)$ and $\alpha_u^+(0) = \alpha_v^+(0)$, while for any $w \neq u, v$ both $\alpha_w^+(0)$ and $\alpha_w^-(0)$ are equal to zero. We will show that, as a consequence of this, whenever $u$ and $v$ will be symmetrical with respect to a symmetry axis $A$ the destructive interference will result in a complete cancellation of the wavefunction amplitude on the nodes of $A$.

We now let $\left|\psi_0^-\right\rangle$ evolve until a time $T$ and we define the average observation probability of the walker at node $v$ as

$$\pi(\alpha_v^-(0))_T = \frac{1}{T}\int_0^T \alpha_v^-(t)\alpha_v^-(t)^* \, \mathrm{d}t \tag{6.15}$$

Similarly, we can define $\pi(\alpha_v^+(0))_T$ given $\left|\psi_0^+\right\rangle$. Here we propose to take the limit of $\pi(\alpha_v^-(0))_T$ and $\pi(\alpha_v^+(0))_T$ as $T \to \infty$, which we simply denote as $\pi(\alpha_v^-(0))$ and $\pi(\alpha_v^-(0))$, respectively. We now show how to compute this limit analytically. Note that here we will refer to a general quantum walk $\left|\psi_t\right\rangle$, but the same observations will hold for $\left|\psi_t^-\right\rangle$ and $\left|\psi_t^+\right\rangle$.

Let $\Phi\Lambda\Phi^\top$ be the spectral decomposition of the graph normalized Laplacian and let $P_\lambda = \sum_{k=1}^{\mu(\lambda)} \phi_{\lambda,k}\phi_{\lambda,k}^\top$ be the projection operator on the subspace spanned by the $\mu(\lambda)$ eigenvectors $\phi_{\lambda,k}$ associated with the eigenvalue $\lambda$ of the graph normalized Laplacian. The evolution operator of the quantum walk can be then expressed in terms of this set of projectors, i.e.,

$$U^t = \sum_{\lambda=1}^{m} e^{-i\lambda t} P_\lambda \tag{6.16}$$

where $m$ denotes the number of unique eigenvalues of the normalized Laplacian. We now introduce the density matrix which describes the ensemble of quantum states $|\psi_t\rangle$, i.e.

$$\rho_T = \frac{1}{T}\int_0^T |\psi_t\rangle\langle\psi_t| \,\mathrm{d}t \tag{6.17}$$

Note that the diagonal of $\rho_T$ can be encoded as a vector with elements $\pi(\alpha_v(0))_T$. If we let $T \to \infty$, we have already proved that the previous equation simplifies to

$$\rho_\infty = \sum_{\lambda=1}^{m} P_\lambda \rho_0 P_\lambda^\top \tag{6.18}$$

In order to compute the diagonal elements of $\rho_\infty$, we start by rewriting the density matrix appearing in Eq. 6.18 as

$$\rho_\infty = \sum_{\lambda=1}^{m} \left( \sum_{k_1=1}^{\mu(\lambda)} \phi_{\lambda,k_1}\phi_{\lambda,k_1}^\top \right) |\psi_0\rangle\langle\psi_0| \left( \sum_{k_2=1}^{\mu(\lambda)} \phi_{\lambda,k_2}\phi_{\lambda,k_2}^\top \right) \tag{6.19}$$

By rearranging the terms of the previous equation we get

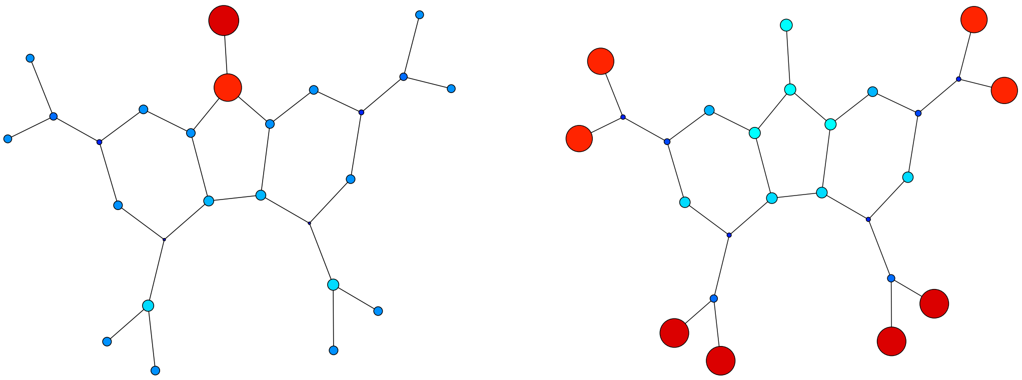
$$\rho_\infty = \sum_{\lambda=1}^{m} \left( \sum_{k_1=1}^{\mu(\lambda)} \phi_{\lambda,k_1}\phi_{\lambda,k_1}^\top \rho_0 \phi_{\lambda,k_1}\phi_{\lambda,k_1}^\top + \sum_{k_1=1}^{\mu(\lambda)} \sum_{k_2\neq k_1}^{\mu(\lambda)} \phi_{\lambda,k_1}\phi_{\lambda,k_2}^\top \rho_0 \phi_{\lambda,k_1}\phi_{\lambda,k_2}^\top \right) \tag{6.20}$$

where $\rho_0 = |\psi_0\rangle\langle\psi_0|$. Note that each term of the inner summations can be written as the product of a scalar by a matrix, and thus we can rewrite the previous equation as

$$\rho_\infty = \sum_{\lambda=1}^{m} \left( \sum_{k_1=1}^{\mu(\lambda)} \left(\phi_{\lambda,k_1}^\top \rho_0 \phi_{\lambda,k_1}\right) \phi_{\lambda,k_1}\phi_{\lambda,k_1}^\top \right. \\ \left. + \sum_{k_1=1}^{\mu(\lambda)} \sum_{k_2\neq k_1}^{\mu(\lambda)} \left(\phi_{\lambda,k_2}^\top \rho_0 \phi_{\lambda,k_1}\right) \phi_{\lambda,k_1}\phi_{\lambda,k_2}^\top \right) \tag{6.21}$$

The $ij$th element of $\rho_\infty$ can then be computed as

$$\rho_\infty(i,j) = \sum_{\lambda=1}^{m} \left( \sum_{k_1=1}^{\mu(\lambda)} \left(\phi_{\lambda,k_1}^\top \rho_0 \phi_{\lambda,k_1}\right) \phi_{\lambda,(i,k_1)}\phi_{\lambda,(j,k_1)}^\top \right. \\ \left. + \sum_{k_1=1}^{\mu(\lambda)} \sum_{k_2\neq k_1}^{\mu(\lambda)} \left(\phi_{\lambda,k_2}^\top \rho_0 \phi_{\lambda,k_1}\right) \phi_{\lambda,(i,k_1)}\phi_{\lambda,(j,k_2)}^\top \right) \tag{6.22}$$



(a) $\tau = 0.01$ (b) $\tau = 0.02$

Figure 6.7: The nodes with degree 1 have a lower probability of being observed compared to the other nodes, and thus they can be accidentally identified as axial. Here we draw each node with a diameter that is proportional to the number of times in which the node has been identified as being axial.

where $\phi_{\lambda,(i,k)}$ denotes the $i$th element of the $k$th eigenvector associated with $\lambda$. Finally, note that if all the eigenvalues of the normalized Laplacian are distinct, the equation for $\pi(\alpha_v(0))$ further reduces to

$$\pi(\alpha_v(0)) = \sum_{\lambda=1}^{m} \phi_\lambda^\top \rho_0 \phi_\lambda \phi_{v,\lambda}^2 = \sum_{\lambda=1}^{m} \left(\langle \psi_0 | \phi_\lambda \rangle\right)^2 \phi_{v,\lambda}^2 \tag{6.23}$$

We now prove that when two nodes $v_1$ and $v_2$ are symmetrical and the initial state of the walk is $|\psi_0^-\rangle$, the average observation probability of the nodes of the symmetry axis will be zero.

**Theorem 6.2.1.** If a pair of nodes $v_1, v_2$ is symmetrical with respect to a symmetry axis $A$ and $\alpha^-_{v_1}(0) = -\alpha^-_{v_2}(0)$, then $\pi(\alpha^-_w(0)) = 0$, $\forall w \in A$.

*Proof.* Recall that

$$\pi(\alpha^-_w(0))_T = \frac{1}{T}\int_0^T \alpha^-_w(t)\alpha^-_w(t)^*\, \mathrm{d}t \tag{6.24}$$

and $\alpha^-_w(0) = \sum_j e^{-iL_{wj}t}\alpha^-_j(0)$, where $L_{wj}$ is the element $(w,j)$ of the graph normalized Laplacian. We now show that, under the hypothesis of the theorem, $\alpha^-_w(t) = 0$ at each instant $t$ and thus it trivially follows that $\pi(\alpha^-_w(0)) = 0$.

First note that according to Eq. 6.14 we can write

$$\alpha_w(0) = \frac{e^{-iL_{wv_1}t} - e^{-iL_{wv_2}t}}{\sqrt{2}} \tag{6.25}$$

and thus we simply need to prove that whenever $v_1$ and $v_2$ are symmetrical with respect to $w$ then $e^{-iL_{wv_1}t} = e^{-iL_{wv_2}t}$. This follows again from Lemma 5.4.1 and noting that, since by hypothesis $w$ is a node of the symmetry axis for $v_1$ and $v_2$, we have that $e^{-iL_{wv_1}t} = e^{-iL_{wv_2}t}$, which concludes the proof. □

Clearly the converse of Theorem 6.2.1 does not hold. In fact, if we were able to prove the converse then we could give a polynomial-time solution to the graph isomorphism problem.

### 6.2.2 Symmetries Detection Criterion

Given the setup we have just described, we can now proceed to the axial nodes identification. Recall once again that, as Fig. 6.7 shows, by observing only the evolution of the anti-phase walk one tends to favor nodes with a low degree, where the average observation probability is usually very low.

We hence propose to identify those vertices that simultaneously show a low observation probability under destructive interference and a high observation probability under constructive interference as axial nodes. In fact, one can easily show along the same lines of Theorem 6.2.1 that when the initial state is $|\psi_0^+\rangle$, if $v_1, v_2$ are symmetrical with respect to $w$ their contributes on $w$ will constructively combine rather than cancel out, and thus the resulting observation probability on $w$ will be higher. With this setting to hand, a node $w$ is identified as being axial if there exists at least one pair of nodes $u$ and $v$ for which

$$\eta_w = \frac{\pi(\alpha_w^+(0))}{\pi(\alpha_w^-(0))} > \tau \tag{6.26}$$

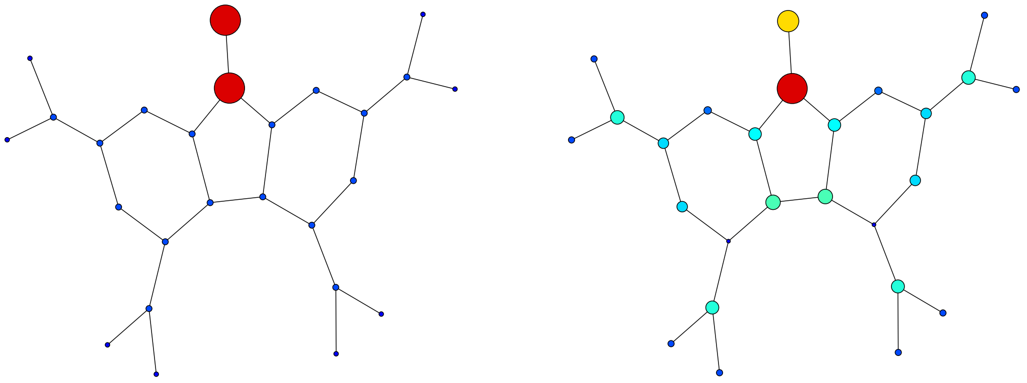
where $\tau$ is a given threshold. Fig. 6.8 clearly shows that with this new criterion we are able to overcome the limitations highlighted in Fig. 6.7.

Note that, according to Theorem 6.2.1, in the case of an exact symmetry $\eta_w = \infty$. In real-world scenarios, however, we have to deal with the presence of structural noise, which will eventually break the symmetries of the graph. We argue that in this case the value of $\eta_w$ can still be used to detect approximate axial symmetries and to characterise the graph structure. Consider for example the toy graph of Fig. 6.9. As we can see, for higher values of $\tau$ we detect the presence of a two-node axis of symmetry. Surprisingly, as we relax the threshold a different pattern is revealed, with 3 nodes rotating around a central axis. This can be explained by observing that the toy graph of Fig. 6.9 is actually a star graph with 3 leaves connected to a central root node. Clearly the root represents the symmetry axis, while the extra edge connecting two of the leaves can be interpreted as structural noise.

The simple procedure described above can be used to establish if two nodes are symmetrical with respect to an axis. In order to detect all the symmetry axis of a graph, one can simply iterate the same procedure for each pair of nodes of the graph. Moreover, given a pair of nodes, we are able to estimate the symmetry axes sizes by counting the number of nodes $w$ where $\eta_w > \tau$.

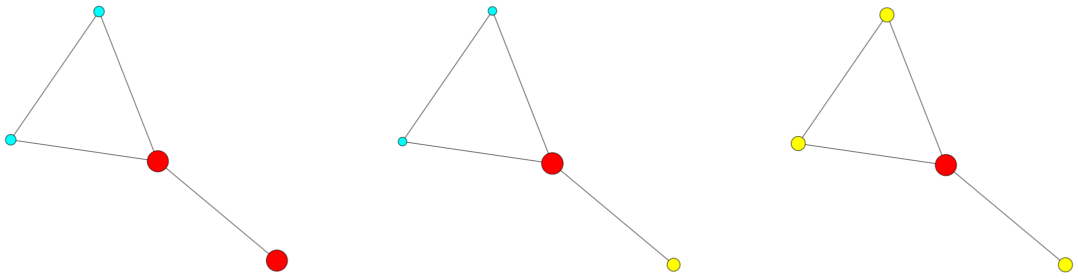
### 6.2.3 Experimental Results

In this Section, we validate the proposed approach by performing a series of experiments on both synthetic data and real-world data. The synthetic data is composed



(a) $\tau = 10$ (b) $\tau = 2$

Figure 6.8: The axial symmetries detected with the algorithm introduced in this Section. Note the importance of taking the evolution of both walks into account.



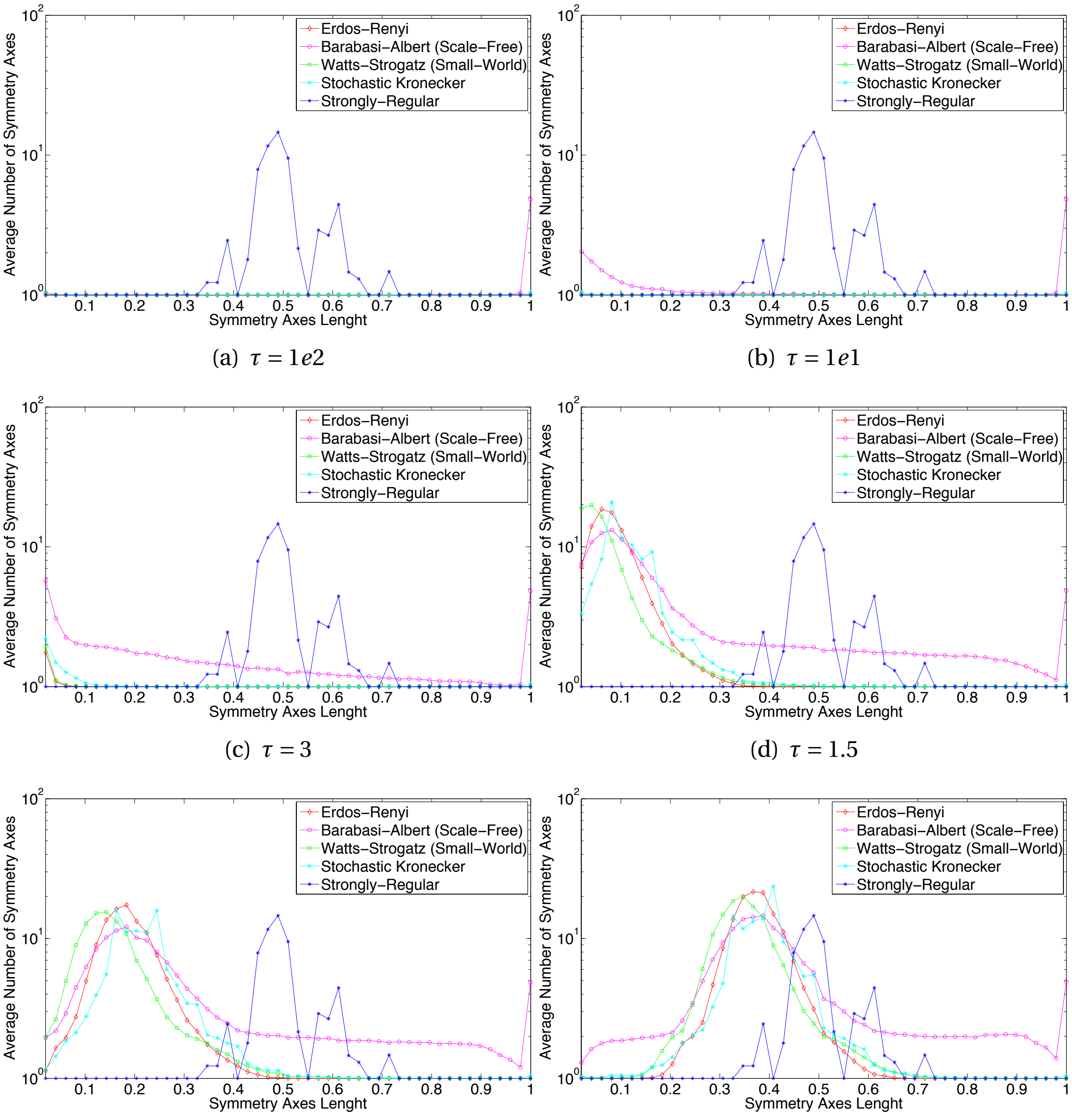
(a) $\tau = 4$ (b) $\tau = 3$ (c) $\tau = 2$

Figure 6.9: A noisy 4 nodes star. As the threshold is relaxed, the original axis of symmetry is revealed.

of Erdös-Rényi random graphs [53], small-world graphs, scale-free graphs, stochastic Kronecker graphs [92] (which exhibit both small-world and scale-free properties), and strongly regular graphs. A regular graph, i.e., a graph where each vertex has degree $k$, is said to be strongly regular if there are two integers $\lambda$ and $\mu$ such that every two adjacent vertices have $\lambda$ common neighbours and every two non-adjacent vertices have $\mu$ common neighbors. We choose strongly regular graphs because they are known to be highly symmetric and this should be reflected in the experimental results. The real-world data, on the other hand, is composed of a set of road networks.

### Synthetic data

For each graph in the dataset, we compute its symmetry axes together with their sizes, as explained in the previous Section. Figure 6.10 shows the distribution of the symmetry axes length for each type of graph, for different choices of the threshold $\tau$. Note that local symmetries correspond to larger axes, since the axis size is equal to the number of nodes of the graph minus the size of the symmetric orbit, which in the case of a local



(a) $\tau = 1e2$

(b) $\tau = 1e1$

(c) $\tau = 3$

(d) $\tau = 1.5$

(e) $\tau = 1.3$

(f) $\tau = 1.1$

Figure 6.10: Symmetry axes distribution. Note that as the threshold varies, the shape of the strongly-regular graphs distribution remains unaltered, as the symmetries present in this category are all exact. Recall that the higher the threshold, the stricter it is.

symmetry is clearly small. On the other hand, a global symmetry will correspond to a smaller symmetry axis. In other words, a left peaked distribution indicates the presence of global symmetries, while a right peaked distribution indicates the presence of local symmetries.

Note that the distribution for the strongly-regular graphs remains unaltered when we change $\tau$. This is because the graphs in this category possess exact symmetries, due

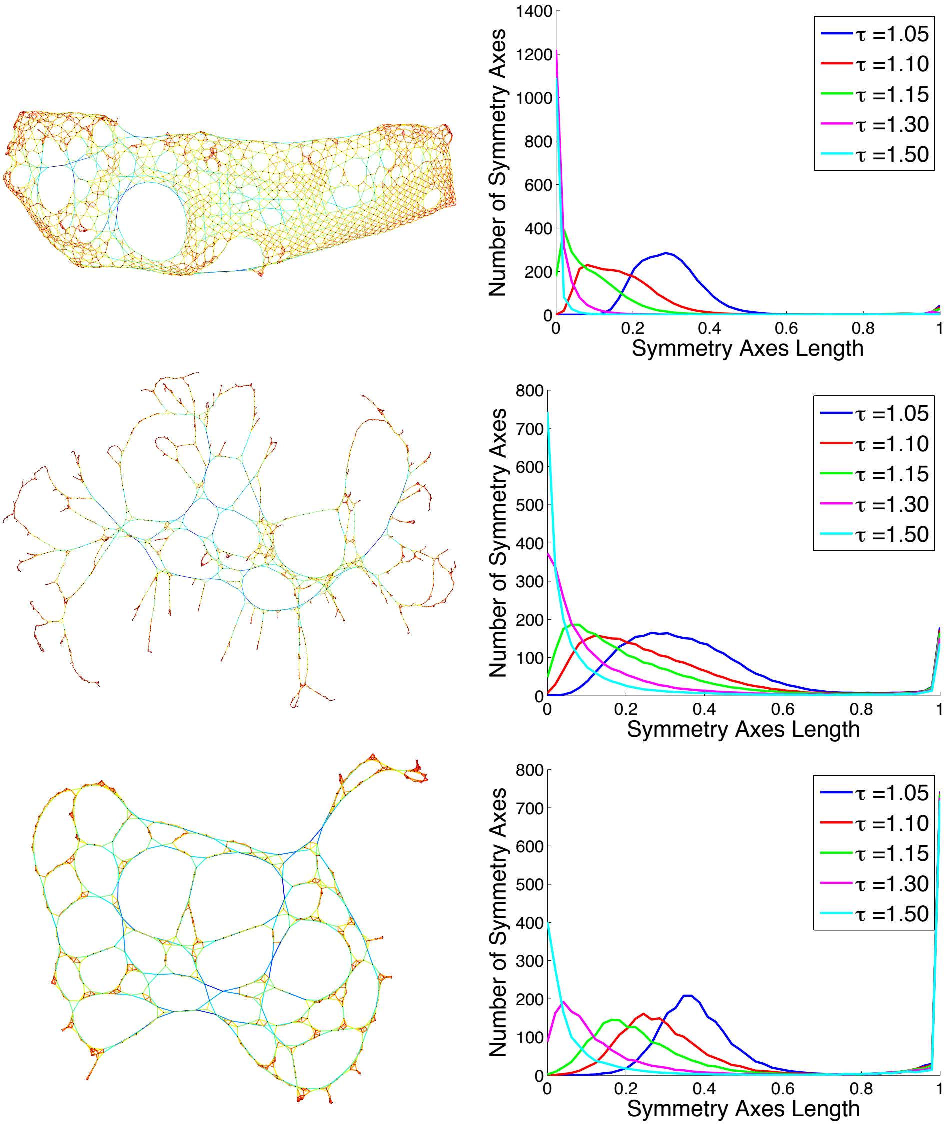


Figure 6.11: Road networks of the cities of Hollywood, Petropolis and Chengkan, along with the corresponding axes length distributions. Note how different layouts give rise to different distributions.

to their regular structure. Hence the probability of the walker being found at a node belonging to a symmetry axis is exactly zero and thus $\eta_w = \infty$ regardless of the threshold value. Note, moreover, that the high number of symmetry axes belonging to this class of graphs is exactly what we would expect given the high degree of symmetry displayed by strongly-regular graphs. As for the other graph models, Figure 6.10 shows that the number of exact symmetries is clearly lower. In particular, we observe the presence of a moderate amount of exact local symmetries in the scale-free graphs, which are probably due to the presence of small trees rooted at a hub node. The class of scale-free

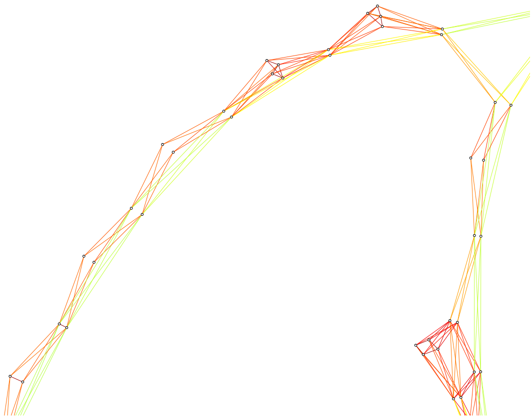


Figure 6.12: A detail of the road network of Chengkan, showing the particular linkage pattern that characterizes this city.

graphs is also that which is most easily separated from the remainder, being characterized by a fat and long tail on the right. The small-world, Erdös-Rényi and Kronecker graphs, on the other hand, show very similar distributions. It is interesting to note that the behaviour of the stochastic Kronecker graphs, which possess both scale-free and small-world properties, seems to be dominated by their small-world behaviour. More generally, Figure 6.10 shows that we can, to some extent, separate graphs belonging to different graph models on the basis of their symmetry axes distributions.

### Real-world data

Road networks are a typical example of technological networks, i.e. man-made networks designed for the distribution of resources. Other examples include power grids, airline routes, river networks and the Internet. In this Section we apply our algorithm to 3 different city layouts, namely a portion of the city of Hollywood in the USA, the city of Petropolis in Brazil and the village of Chengkan in China. Each city is represented by an undirected graph which is the dual of its road network, i.e., each node is a street and two nodes are connected by an edge if they meet at a crossing. The sizes of the resulting graphs is 1991 nodes for Hollywood, 1969 for Petropolis and 1272 for Chengkan. For each graph, we compute the approximate symmetry axes and their length for different thresholds.

Figure 6.11 shows the embeddings of the three cities and the corresponding distributions. We observe that different layouts of the cities give rise to different distributions. As expected, the first city, which shows a very regular grid-like structure, responds markedly to the presence of approximate global symmetry, and shows little or no local symmetry. On the other hand, the second city displays for each threshold a wider distribution, and it seems to possess a number of exact local symmetries which are reflected in the far right side of the plot. A similar pattern was displayed by the scale-free graph model in Fig. 6.10. In fact, a visual inspection of the graph confirms the presence of several small hubs, which are typical of scale-free models. Finally, the third city shows a remarkably large number of local symmetries, which arise as a consequence of its very particular linkage pattern, which is shown in Fig. 6.12.

## 6.3 A Quantum Measure of Vertex Centrality

Inspired by the symmetry analysis of the previous Sections, we would like to now shift the focus from the characterization of the whole graph to that of a single node. In particular, we would like to measure the *centrality* of a vertex as the number of times in which it belongs to an axis of symmetry. Establishing the importance of the vertices of a graph is of key importance in the analysis of complex networks, and a number of centrality indices have been introduced in the literature [54]. The most common examples are probably the degree, closeness and betweeness centrality [57, 58, 104]. Each of these measures capture different but equally significant aspects of a vertex importance. The degree centrality naturally interprets the number of edges incident on a vertex as a measure of its “popularity". The closeness centrality links the importance of a vertex to its proximity to the remaining vertices of the graph. Finally, the betweeness centrality is a measure of the extent to which a vertex lies on the paths between others.

In this Section we would like to extend the existing centrality indices by using the continuous-time quantum walk as a means to measure the centrality of a node. Given the symmetry analysis framework developed in the previous Section, a first guess could be that of measuring the centrality of a vertex as the number of times in which it belongs to an axis of symmetry. However, in this case computing the centrality of a single vertex would require iterating the symmetries detection over all the pairs of nodes of the graph. Thus, we propose an alternative measure which relates the importance of a vertex to the influence that its initial phase has on the evolution of a suitably defined quantum walk.

### 6.3.1 QJSD Centrality

In order to measure the centrality of vertex $v$, we define two quantum walks where $v$ is initially set to be in phase and in antiphase with the respect to the other nodes, respectively. That is, we define two walks $\left|\psi_0^{v-}\right\rangle = \sum_{u\in V} \alpha_u^{v-}(0)\left|u\right\rangle$ and $\left|\psi_0^{v+}\right\rangle = \sum_{u\in V} \alpha_u^{v+}(0)\left|u\right\rangle$ on $G$ with starting states

$$\alpha_j^{v-}(0) = \begin{cases} -\frac{1}{C} \text{ if } j = v \\ +\frac{1}{C} \text{ otherwise} \end{cases} \qquad \alpha_j^{v+}(0) = \left\{ +\frac{1}{C}\ \forall j \right. \tag{6.27}$$

where $C$ is the normalisation constant such that probabilities sum to 1. Alternatively, we may define the initial amplitude to be proportional to the square root of the nodes degree, i.e.,

$$\widetilde{\alpha}_j^{v-}(0) = \begin{cases} -\frac{\sqrt{d}_j}{C} \text{ if } j = v \\ +\frac{\sqrt{d}_j}{C} \text{ otherwise} \end{cases} \qquad \widetilde{\alpha}_j^{v+}(0) = \left\{ +\frac{\sqrt{d}_j}{C}\ \forall j \right. \tag{6.28}$$

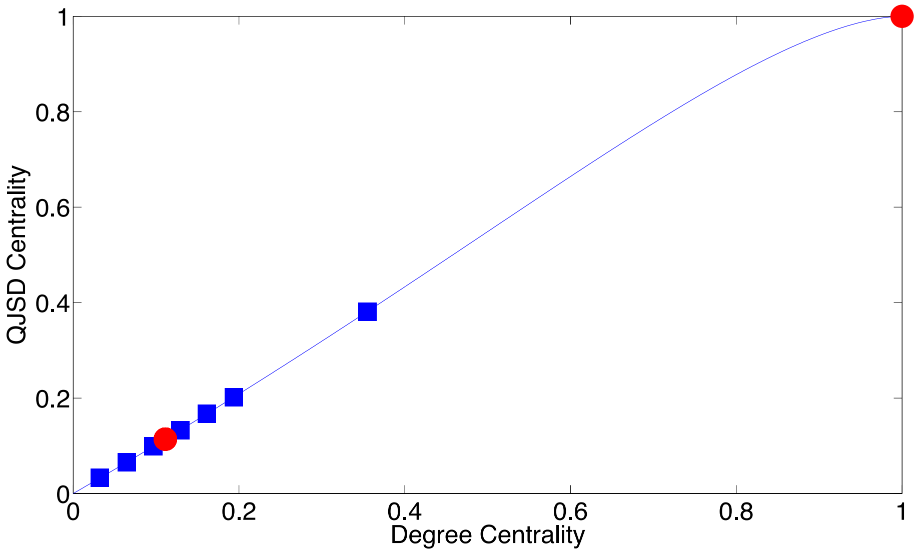


Figure 6.13: The correlation between degree and QJSD centrality, for a star graph (red dots) and a scale-free graph (blue squares). The blue line shows the predicted dependency between the two centrality indices.

Now let $\rho_v$ and $\sigma_v$ be the density operators which describe the ensembles of quantum states $|\psi_t^{v-}\rangle$ and $|\psi_t^{v+}\rangle$ respectively, i.e.,

$$\rho_v = \lim_{T\to\infty} \frac{1}{T}\int_0^T |\psi_t^{v-}\rangle\langle\psi_t^{v-}|\,\mathrm{d}t \qquad \sigma_v = \lim_{T\to\infty} \frac{1}{T}\int_0^T |\psi_t^{v+}\rangle\langle\psi_t^{v+}|\,\mathrm{d}t \tag{6.29}$$

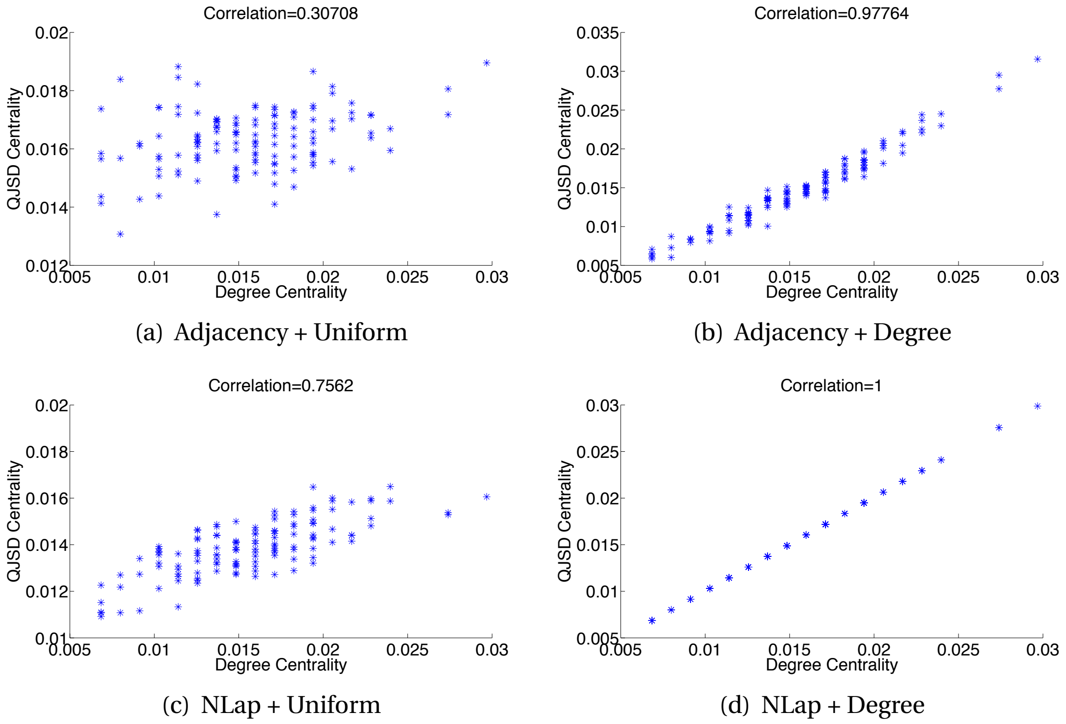
Then we can measure how the initial phase of the vertex $v$ affects the evolution of the quantum walks by computing the distance between the quantum states defined by $\rho_v$ and $\sigma_v$, i.e.,

$$C_{JS}(v) = D_{JS}(\rho_v, \sigma_v) \tag{6.30}$$

We stress that the computation of the QJSD centrality is entirely based on principled observables. As a consequence, it should be possible, at least in theory, to design a quantum algorithm to compute the QJSD centrality that could benefit from the power of quantum computers. However, the design of such an algorithm is clearly beyond the scope of this thesis.

We are now interested in studying to what extent the QJSD centrality depends on the degree of the nodes. It is known, in fact, that the classical versions of centrality, like the betweeness centrality, are highly correlated with the degree. Let the initial states of the walks be defined as in Eq. 6.28, and let the normalized Laplacian be the Hamiltonian of our system. We start by observing that $|\widetilde{\psi}_0^{v+}\rangle = \sum_{u\in V}\widetilde{\alpha}_u^{v+}(0)\,|u\rangle$ corresponds to the eigenvector $\phi_0$ associated to the zero eigenvalue of the Hamiltonian, and as a consequence $|\widetilde{\psi}_0^{v+}\rangle$ will remain constant over time. In other words, we have that

$$\sigma_v = |\widetilde{\psi}_0^{v+}\rangle\langle\widetilde{\psi}_0^{v+}| \tag{6.31}$$



(a) Adjacency + Uniform

(b) Adjacency + Degree

(c) NLap + Uniform

(d) NLap + Degree

Figure 6.14: Correlation between the QJSD centrality and the degree centrality for different choices of the Hamiltonian (adjacency matrix or normalized Laplacian) and of the initial state (normalized uniform distribution or normalized degree distribution). Interestingly, when we choose the normalized Laplacian as the Hamiltonian and the amplitudes are initialised according to the degree distribution, the correlation is almost linear.

From this, it immediately follows that the spectrum of $\sigma_v$ is composed of a single eigenvector $\phi_0$ with eigenvalue equal to 1. As a consequence of this and of Equation 6.16, $\rho_v$ and $\sigma_v$ are simultaneously diagonalizable, and therefore each eigenvalue of their sum is a sum of eigenvalues of $\rho_v$ and $\sigma_v$. More precisely, when the two walks are initialised as in Eq. 6.14, all the eigenvalues $\mu_i$ of $\frac{\rho_v+\sigma_v}{2}$ will be equal to the eigenvalues of $\rho_v$, except for the eigenvalue $\mu_0 + 1$ which is associated to the common eigenvector $\phi_0$. We now show that, as a consequence of this, the QJSD centrality is proportional to the degree centrality. Recall that

$$C_{JS}(v) = D_{JS}(\rho_v, \sigma_v) = H_N\left(\frac{\rho_v + \sigma_v}{2}\right) - \frac{1}{2}\left(H_N(\sigma_v) + H_N(\rho_v)\right) \tag{6.32}$$

From Equations 6.14 and 6.16 we have that $H_N(\sigma_v) = 0$. As a consequence,

$$\begin{aligned}
D_{JS}(\rho_v, \sigma_v) &= H_N\left(\frac{\rho_v + \sigma_v}{2}\right) - \frac{1}{2}H_N(\rho_v) \\
&= -\frac{\mu_0 + 1}{2}\log_2\frac{\mu_0 + 1}{2} - \sum_{i \neq 0}\frac{\mu_i}{2}\log_2\frac{\mu_i}{2} + \frac{1}{2}\sum_i \mu_i \log_2 \mu_i \\
&= \frac{\mu_0 + 1}{2} - \frac{\mu_0 + 1}{2}\log_2(\mu_0 + 1) + \sum_{i \neq 0}\frac{\mu_i}{2} - \frac{1}{2}\sum_{i \neq 0}\mu_i \log_2 \mu_i + \frac{1}{2}\sum_i \mu_i \log_2 \mu_i \\
&= 1 - \frac{1}{2}\log_2(\mu_0 + 1) + \frac{\mu_0}{2}\log_2\frac{\mu_0}{\mu_0 + 1} \qquad (6.33)
\end{aligned}$$

where $\mu_i$ denotes the $i$th eigenvalue of $\rho_v$ and we used the fact that $\sum_i \mu_i = 1$. We now proceed to show that $\mu_0$ is proportional to the degree of node $v$, and therefore the QJSD centrality is proportional to the degree centrality. In fact, we have that

$$\mu_0 = \langle \phi_0 | \rho_0 | \phi_0 \rangle = \langle \phi_0 | \widetilde{\psi}_0^{v-} \rangle^2 = \left(1 - \frac{d_v}{|E|}\right)^2 \qquad (6.34)$$

where $d_v$ is the degree of $v$ and $|E|$ denotes the number of edges in the graph. In other words, when we take the normalized Laplacian as our Hamiltonian and we initialise the walks according to Eq. 6.28, the QJSD centrality turns out to be quasi-linearly correlated with the degree centrality. Fig. 6.13 shows the correlation between the QJSD centrality and the degree centrality for a scale-free random graph and a star graph. In the case of a general graph, the two measures appear to be exactly linearly correlated, which explains the behaviour observed in Fig.6.14. The non-linear behaviour, in fact, is observed only for nodes with a normalized degree close to 1, as in the case of a star graph.

Note that, although so far we assumed that the Hamiltonian of the quantum walk was the graph normalized Laplacian, the Laplacian and the adjacency matrix have also been used in the literature. However, the evolution of the walk and thus the QJSD centrality can vary a lot under these different settings. Fig. 6.14 shows the correlation between the QJSD centrality and the degree centrality computed on a stochastic Kronecker graph for different choices of the initial state and the Hamiltonian. As we can see, the correlation is close to 1 when Eq. 6.28 is used to define the initial amplitude. Such a strong correlation seems to imply that this variant of our measure is useless, as the degree centrality is certainly much easier to calculate. However, for a number of vertices of the graph the order which results from the two measures is actually different, and it is in these small differences that lies the significance of our index.

In the remainder of this Chapter we will use the adjacency matrix as the Hamiltonian and we will set the initial state according to Eq. 6.28, as the high correlation with the degree centrality makes the QJSD centrality fairly easy to interpret. Finally, we conclude this Section with an evaluation of the impact of the magnitude of the initial amplitude on the nodes centrality. More formally, we define the starting states with

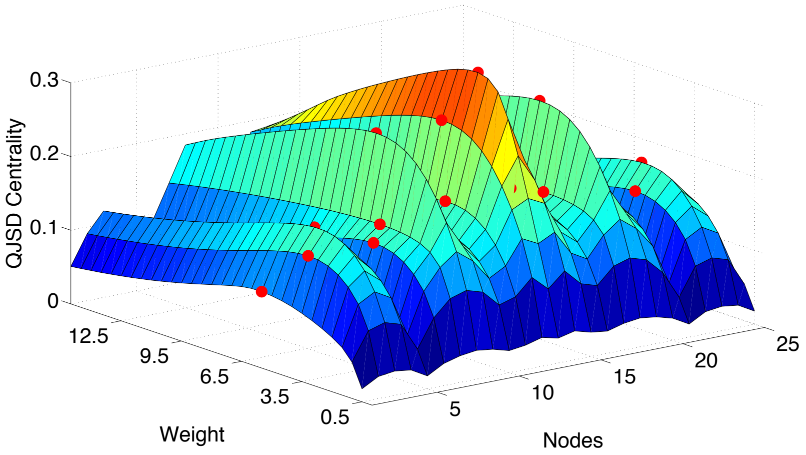
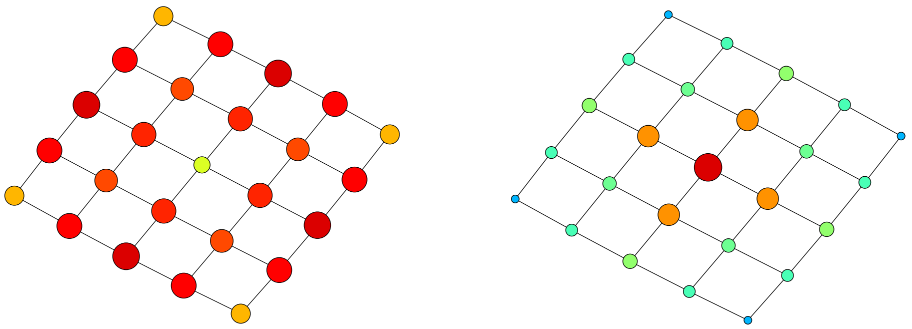


Figure 6.15: The QJSD centrality as a function of the nodes and of the weight $w$. For each node, the highest centrality corresponds to the choice of a different weight (marked with a red dot). Moreover, the order of the nodes varies as $w$ varies.



(a) Original

(b) Modified

Figure 6.16: The QJSD centrality for a 5x5 mesh. The left and rigth figures show the resulting QJSD centrality when the walk is initialised as in Eq. 6.28 and in Eq. 6.35 respectively.

amplitudes

$$\tilde{\alpha}_j^{v-}(0) = \begin{cases} -\frac{w\sqrt{d_j}}{C} \text{ if } j = v \\ +\frac{\sqrt{d_j}}{C} \text{ otherwise} \end{cases} \qquad \tilde{\alpha}_j^{v+}(0) = \begin{cases} +\frac{w\sqrt{d_j}}{C} \text{ if } j = v \\ +\frac{\sqrt{d_j}}{C} \text{ otherwise} \end{cases} \tag{6.35}$$

where $w$ is a real value which is used to change the initial magnitude of the wavefunction on $v$. The idea is that the higher the initial magnitude on $v$, the more marked will the interference effects be. Fig. 6.15 shows the value of the QJSD centrality for the nodes of a $5 \times 5$ mesh as $w$ is increased from 0.1 to 15. Note that each node achieves its maximum centrality for a different choice of $w$ and that for different weights the order of the nodes induced by the QJSD centrality varies. Ideally, given a node $v$, one may want to set the weight so as to maximize the divergence for that specific node. However, since iterating the measurement of the QJSD centrality for different weights may prove too cumbersome, we propose to set the amplitude of $v$ equal to the sum of the amplitudes on the remaining nodes. Although this is clearly not the optimal solution, as Fig. 6.16 shows the proposed weighting scheme yields a result which is certainly closer to common sense than the original scheme. Therefore, in the remainder of this Chapter we will assume that the initial amplitude of the walk is defined as in Eq. 6.35.

### 6.3.2 Experimental Results

We apply the QJSD centrality to two commonly used network datasets, namely Zachary's karate club [151] and Padgett's network of marriages between the 16 most eminent Florentine families in the 15th century [107]. Fig. 6.17 shows Zachary's karate club network, where each vertex is drawn with a diameter that is proportional to the QJSD centrality. We see that there are two main actors, node #1 and node #2, which correspond to the instructor and the administrator of the club. Note that using our measure the administrator turns out to be the node with the highest centrality, which is also the most central according to the betweeness centrality, while the degree centrality elects the instructor as the most important node. However, the betweeness centrality indicates as the second most important actor node #3, as this vertex has many contacts with both the members of administrator cluster and the members of instructor cluster and thus it is misunderstood as a center by the betweenness centrality. Finally, node #4 is identified as the third most important by the degree centrality, leaving node #3 at the fourth place, although the latter is more central in the sense that it shares many links with nodes of the administrator group and of the instructor group.

Padgett's network of marriages is depicted in Fig. 6.18. In Table 6.1, we show the

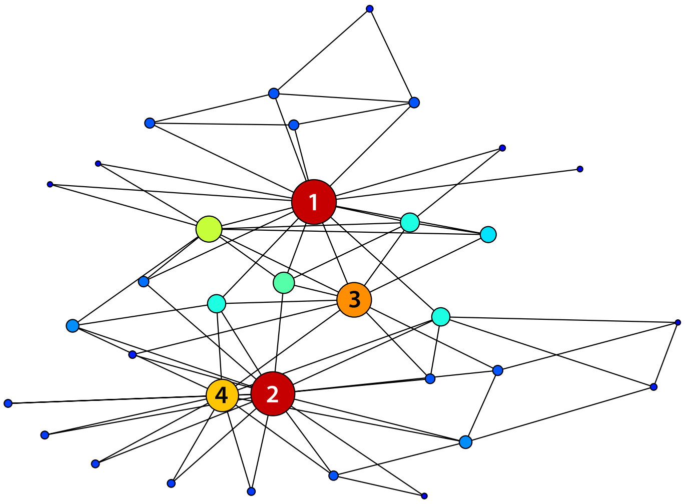


Figure 6.17: Zachary's karate club network, where we have drawn each node with a diameter that is proportional to its QJSD centrality.

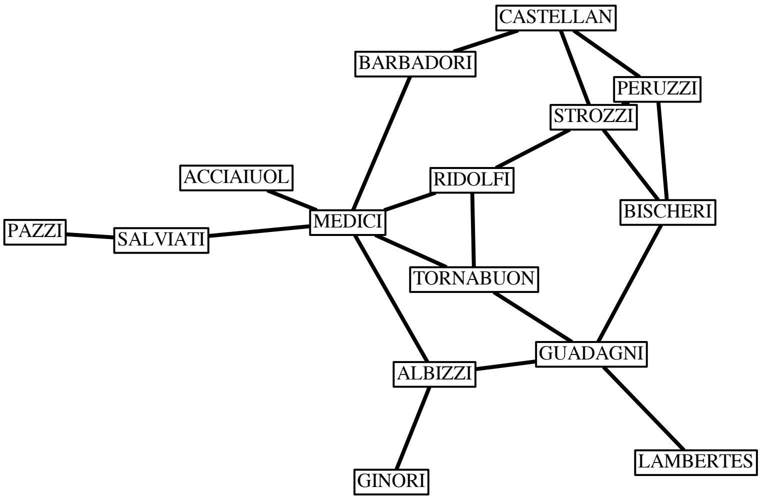


Figure 6.18: Padgett's network of marriages between eminent Florentine families in the 15th century [107]. We omit the Pucci, which had no marriage ties with other families.

ranking of the 15 families according to their QJSD centrality. As expected, the Medici easily best the Strozzi, which are their main rivals, which agrees with the idea that Medici's supremacy was largely due to their skills in manipulating the marriage network. Interestingly the Pazzi, which is the most loosely connected family of the graph, achieves the lowest centrality. Moreover, Peruzzi, Castellan and Bischeri all get a higher centrality than Albizzi, although the degree of the four vertices is the same. This fits nicely with the fact that the Peruzzi, Castellan, Bischeri and Strozzi form a rather cliquey group in which the actors support each other, while the Albizzi remains a bit more isolated.

| Family | Centrality | Family | Centrality |
|---|---|---|---|
| Medici | 0.3120 | Albizzi | 0.1299 |
| Strozzi | 0.2103 | Barbadori | 0.0913 |
| Guadagni | 0.1831 | Salviati | 0.0697 |
| Peruzzi | 0.1531 | Ginori | 0.0421 |
| Castellan | 0.1516 | Acciaiuol | 0.0389 |
| Ridolfi | 0.1491 | Lambertes | 0.0225 |
| Bischeri | 0.1440 | Pazzi | 0.0221 |
| Tornabuoni | 0.1410 | | |

Table 6.1: The QJSD centrality of the families of Padgett's network [107].

## 6.4 Conclusions

Much recent research in the quantum walks domain has shown the existence of a link between the interesting properties shown by quantum walks on graphs and the presence of symmetrical motifs in the graphs structure. This particular structure, in fact, can lead to remarkable interference effects, both constructive and destructive. In this Section 6.1 we have proposed a way to measure the presence of symmetries in a graph using the quantum Jensen-Shannon divergence. This in turn has allowed us to design an experiment to analyze the behaviour of the quantum walk without causing the wave function collapse. We showed how to define two mixed states based on two different quantum walks on the graph, and we used the resulting density operators to measure the distance between the two quantum states. In particular, we proved that when the graph possesses a symmetry, the QJSD between the two quantum states is maximum. Our experiments show that a simple measure such as the average of the QJSD matrix is able to capture the structural difference between a symmetrical graph and an Erdös-Rényi random graph, even in the presence of moderate Erdös-Rényi noise, as well as to distinguish between different random graph models. In Section 6.2 we have shown how to explicitly detect approximate axial symmetries by performing a semi-classical analysis of the interference. We demonstrated the efficacy of our approach by analyzing both synthetic and real-world data. Finally, in an attempt to relate the importance of a vertex to its influence on the interference patterns emerging during the quantum walk evolution, in Section 6.3 we have proposed to use of the quantum Jensen-Shannon divergence between two suitably defined quantum states to introduce a novel centrality measure.

# 7
# Conclusions

In this thesis we introduced a wide spectrum of techniques for the modeling, classification and analysis of graph structures. Our contributions can be identified into four different areas. Chapter 3 introduced a novel algorithm for the extraction of medial surfaces (3D skeletons) from three-dimensional shapes. The extraction of skeletons is a common pre-processing technique in the analysis of 2D shapes. With the skeleton to hand, one can segment it into different components and use a graph to represent the relation between these parts. In this sense, our medial surface extraction algorithm represents a first vital step in the pipeline for acquiring and analyzing 3D shapes. Chapter 4 and Chapter 5 dealt with the classification of graphs, using generative and discriminative approaches, respectively. More specifically, we introduced a novel algorithm for learning a generative model for graphs in Chapter 4, together with a novel information-theoretic criterion for model selection. In Chapter 5, on the other hand, we described a new kernel for unattributed and attributed graphs which is based on a quantum-mechanical analysis of the graph structure. In the same Chapter, we proposed a way to increase the performance of the kernel by applying standard manifold learning techniques on it. Finally, in Chapter 6 we used a similar quantum-mechanical framework for analyzing the structure of the graph, with particular attention on the discovery of approximate axial symmetries. We now recap in detail the contributions of this thesis and the future directions of research.

## Contributions and Novelty

The problem of medial surfaces extraction was addressed in Chapter 3. Although there exists a large number of successful algorithms for the extraction of skeletons from 2D shapes, the addition of a third dimension makes the task of medial surfaces extraction particularly challenging. To this end, we generalized to three dimensions the density-corrected analysis of Torsello and Hancock [138] where we iteratively refined an initial coarse discretization of the shape interior by focusing on those point that were more likely to be skeletal. More precisely, at each iteration we computed the gradient and Laplacian of the distance map, we integrated the log-density in the voxels with a full neighborhood and we alternated thinning and dilation steps to detect skeletal voxels at the current level of resolution. In order to ensure that the original object topology was maintained throughout the process, we adopted the strategy proposed by Malandain et al. [95], which allowed us to efficiently identify which voxels could be removed by exploring the connectivity of their neighborhood. The experimental part clearly demonstrated that our method is efficiently able to recover the medial surface and shows an increased robustness when compared to alternative approaches in the literature. Moreover, we designed a simple alignment procedure to correct the displacement of the extracted skeleton with respect to the true underlying medial surface. This is an issue that stems from the voxelization of the shape itself, and to our knowledge the proposed approach is novel.

Chapter 4 introduced the problem of learning a generative model which is able to capture the relations and observation probabilities of the nodes of a set of observed graphs. In order to describe the structural variations of the training set, we made the naïve assumption that the observation of each node and each edge was independent of the others, but we allowed correlations to pop up by actually learning a mixture of models. When learning a structural model, a common mistake is that of assuming a maximum likelihood estimation, or simply a single estimation for the set of node correspondences. We showed how to eliminate the bias resulting from a single estimation by averaging over the set of all possible correspondences. Given the super-exponential growth of this set, however, we decided to approximate the computation using an importance sample strategy to select a limited number of correspondences. Finally, we adopted a classical MML approach to penalize complex models and select which mixture components and nodes required pruning. In addition to this, later in the Chapter we proposed a novel information-theoretic framework for model selection which relied on the maximization of the capacity of a suitably defined communication channel.

Although the generative model of Chapter 4 proved effective in a number of computer vision related classification tasks, due to the complexity of efficiently sampling the hidden correspondences and estimating the observation probabilities, its use is restricted to graphs with a limited number of nodes. Moreover, it is known that generally, although lacking the flexibility of generative approaches, deterministic approaches can lead to higher classification performances. For these reasons, in Chapter 5 we introduced a novel graph kernel which works both on unattributed and attributed graphs. After observing that the dynamics of quantum walks on graphs are greatly influenced by the presence of symmetrical structures, we designed a simple yet effective way to measure the similarity between two graphs. In particular, we proposed to let two suitably defined continuous-time quantum walks evolve on a union of the two graphs, and we computed the divergence between the respective quantum states. To this end, we made use of the quantum Jensen-Shannon divergence, a measure which has recently been introduced as a means to compute the distance between quantum states [94, 86]. Although we were unable to prove the positive definiteness of this kernel, we carried out an extensive experimental evaluation to show that our kernel can easily outperform alternative graph kernels. We also proposed a way to enhance the performance of the kernel by computing a low-dimensional embedding where the different classes are better separated. The idea stemmed from the observation that the multidimensional scaling embeddings on this kernel showed a strong horseshoe shape distribution, a pattern which is known to arise when long range distances are not estimated accurately. We hence proposed to use Isomap to embed the graphs using only local distance information onto a new vectorial space with a higher class separability and carried out an extensive experimental evaluation to show the effectiveness of the approach.

The final Chapter built on the quantum-mechanical framework introduced in Chapter 5 to develop a set of novel algorithms for the analysis of graph structure. Given the close connection between structural symmetries and destructive (constructive) interference of quantum walks, we decided to design a simple algorithm to measure the degree of approximate axial symmetries possessed by a graph. Not only our approach is completely novel, but this also turns out to be an extremely hard task, since approximate symmetries are by definition hard to characterize. In other words, it is not clear how to enumerate the number of approximate symmetries of a graph, and thus it is difficult if not impossible to establish a ground truth. However, we designed a series of experiments to carefully evaluate the properties of our algorithm, and we showed that it is indeed able to capture the structural difference between a symmetrical graph and an Erdös-Rényi random graph, even in the presence of moderate Erdös-Rényi noise, as well as to distinguish between different random graph models. Moreover, we proved that when the graph possesses a symmetry, our measure, which is based again on the quantum Jensen-Shannon divergence, achieves its maximum value. We then proposed a way to explicitly detect approximate axial symmetries by performing a semi-classical analysis of the quantum walk interference and we tested our approach both on synthetic and real-world data. Finally, in an attempt to relate the importance of a vertex to its influence on the interference patterns emerging during the quantum walk evolution, in last part of Chapter 6 we proposed a novel node centrality measure which is once again based on evaluating the quantum Jensen-Shannon divergence between two suitably defined quantum states.

## Future Work

There are a number of open research questions that were not addressed in this thesis, and will be the subject of future work. Given the skeleton of a 3D object, rather than describing the adjacency relation between the medial sheets in terms of an undirected graphs, one may use a richer structure such as a hypergraph, i.e., a generalization of a graph where a hyperedge can contain an arbitrary number of nodes. In this case, a hyperedge would naturally encode the adjacency relation between a set of intersecting medial sheets. Alternatively, one may adopt a medial scaffold representation as in Chang et al. [39, 40].

The generative model of Chapter 4 could be extended to learn the occurrence of repeating substructures in a graph. In this setting, the input would be a single large graph, or a few instances of it, where a given module that we intend to learn is repeated a number of times, with possible structural variations. This clearly requires rendering the sample one-to-many such that the same model node can map to multiple nodes in the graph, or on multiple graphs. Moreover, the external node should now account for all those graph nodes which are not part of the substructure that we intend to model, rather than simply noise. Thus, the model should be redefined in order to avoid penalizing external nodes too much. Note also that a critical step would be the initial assignment estimation, as we would need a way to estimate the location of all the occurrences of the model in the larger graph.

Finally, Chapter 5 leaves a lot of room for improvement and further study. In this thesis we were unable to prove the positive semidefiniteness of the QJSD kernel, however we observed that both empirical and theoretical evidences suggest that it might be. Moreover, it would be interesting to study in a more systematical way the role of the time parameter. In fact, while we proposed to let $T \to \infty$, we also noted that on a synthetic dataset the best classification accuracy was achieved for a finite value of $T$. To conclude, we should explore the possibility of applying alternative and more sophisticated manifold learning techniques on the kernel. It is known, in fact, that Isomap suffers from several shortcomings, so further work should focus on experimenting with more robust manifold learning techniques.

# Bibliography

[13] ABEYSINGHE, S. S., JU, T., CHIU, W., AND BAKER, M. Shape modeling and matching in identifying protein structure from low-resolution images. In *ACM Symposium on Solid and Physical Modeling: Proceedings of the 2007 ACM symposium on Solid and physical modeling* (2007), vol. 4, pp. 223–232.

[14] AKAIKE, H. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on 19*, 6 (1974), 716–723.

[15] AMBAINIS, A. Quantum walks and their algorithmic applications. *International Journal of Quantum Information 1*, 04 (2003), 507–518.

[16] ARCELLI, C., DI BAJA, G. S., AND SERINO, L. Distance-driven skeletonization in voxel images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 4 (2011), 709–720.

[17] AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. Skeleton extraction by mesh contraction. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 44.

[18] BABAI, L., ERDO˝ S, P., AND SELKOW, S. M. Random graph isomorphism. *SIAM Journal on Computing 9*, 3 (1980), 628–635.

[19] BAERENTZEN, J. A., AND AANAES, H. Signed distance computation using the angle weighted pseudonormal. *Visualization and Computer Graphics, IEEE Transactions on 11*, 3 (2005), 243–253.

[20] BAI, L., AND HANCOCK, E. Graph kernels from the Jensen-Shannon divergence. *Journal of Mathematical Imaging and Vision* (2012), 1–10.

[21] BAI, Y., HAN, X., AND PRINCE, J. L. Digital topology on adaptive octree grids. *Journal of mathematical imaging and vision 34*, 2 (2009), 165–184.

[22] BARABÁSI, A., AND ALBERT, R. Emergence of scaling in random networks. *science 286*, 5439 (1999), 509–512.

[23] BARBER, C. B., DOBKIN, D. P., AND HUHDANPAA, H. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS) 22*, 4 (1996), 469–483.

[24] BARROW, H. G., AND BURSTALL, R. M. Subgraph isomorphism, matching relational structures and maximal cliques. *Inf. Process. Lett. 4*, 4 (1976), 83–84.

[25] Beichl, I., and Sullivan, F. Approximating the permanent via importance sampling with application to the dimer covering problem. *Journal of computational Physics 149*, 1 (1999), 128–147.

[26] Bertrand, G. A parallel thinning algorithm for medial surfaces. *Pattern Recognition Letters 16*, 9 (1995), 979–986.

[27] Biggs, N. *Algebraic graph theory.* Cambridge University Press, 1993.

[28] Bloomenthal, J. Medial-based vertex deformation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), ACM, pp. 147–151.

[29] Blum, H., et al. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form 19*, 5 (1967), 362–380.

[30] Bonev, B., Escolano, F., Lozano, M. A., Suau, P., Cazorla, M. A., and Aguilar, W. Constellations and the unsupervised learning of graphs. In *Graph-Based Representations in Pattern Recognition.* Springer, 2007, pp. 340–350.

[31] Borgwardt, K., and Kriegel, H. Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on* (2005), IEEE, pp. 8–pp.

[32] Briët, J., and Harremoës, P. Properties of classical and quantum jensen-shannon divergence. *Physical review A 79*, 5 (2009), 052311.

[33] Bronstein, A., Bronstein, M., Castellani, U., Dubrovina, A., Guibas, L., Horaud, R., Kimmel, R., Knossow, D., Von Lavante, E., Mateus, D., et al. Shrec 2010: robust correspondence benchmark. In *Eurographics Workshop on 3D Object Retrieval (3DOR'10)* (2010).

[34] Buhmann, J. M. Information theoretic model validation for clustering. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on* (2010), IEEE, pp. 1398–1402.

[35] Buhmann, J. M., Chehreghani, M. H., Frank, M., Streich, A. P., Buhmann, J. M., and Buhmann, J. M. *Information theoretic model selection for pattern analysis.* Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2011.

[36] Bunke, H., Foggia, P., Guidobaldi, C., and Vento, M. Graph clustering using the weighted minimum common supergraph. In *Graph based representations in pattern recognition.* Springer, 2003, pp. 235–246.

[37] Bunke, H., and Shearer, K. A graph distance metric based on the maximal common subgraph. *Pattern recognition letters 19*, 3 (1998), 255–259.

[38] BURES, D. An extension of kakutani's theorem on infinite product measures to the tensor product of semifinite w*-algebras. *Transactions of the American Mathematical Society 135* (1969), 199–212.

[39] CHANG, M.-C., AND KIMIA, B. B. Regularizing 3d medial axis using medial scaffold transforms. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[40] CHANG, M.-C., LEYMARIE, F. F., AND KIMIA, B. B. Surface reconstruction from point clouds by transforming the medial scaffold. *Computer Vision and Image Understanding 113,* 11 (2009), 1130–1146.

[41] CHANG, S. Extracting skeletons from distance maps. *International Journal of Computer Science and Network Security 7,* 7 (2007), 213–219.

[42] CHILDS, A. Universal computation by quantum walk. *Physical review letters 102,* 18 (2009), 180501.

[43] CHUNG, F. R. *Spectral graph theory,* vol. 92. AMS Bookstore, 1997.

[44] COLLINS, M., AND DUFFY, N. Convolution kernels for natural language. In *Advances in neural information processing systems* (2001), pp. 625–632.

[45] CORNEA, N. D., SILVER, D., AND MIN, P. Curve-skeleton properties, applications, and algorithms. *Visualization and Computer Graphics, IEEE Transactions on 13,* 3 (2007), 530–548.

[46] COUR, T., SRINIVASAN, P., AND SHI, J. Balanced graph matching. *Advances in Neural Information Processing Systems 19* (2007), 313.

[47] CRANK, J., AND NICOLSON, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society* (1947), vol. 43, Cambridge Univ Press, pp. 50–67.

[48] CZAJA, W., AND EHLER, M. Schroedinger eigenmaps for the analysis of biomedical data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35,* 5 (2013), 1274–1280.

[49] DEHMER, M. Information processing in complex networks: Graph entropy and information functionals. *Applied Mathematics and Computation 201,* 1 (2008), 82–94.

[50] DENG, W., IYENGAR, S. S., AND BRENER, N. E. A fast parallel thinning algorithm for the binary image skeletonization. *International Journal of High Performance Computing Applications 14,* 1 (2000), 65–81.

[51] EMMS, D., WILSON, R., AND HANCOCK, E. Graph embedding using quantum commute times. *Graph-Based Representations in Pattern Recognition* (2007), 371–382.

[52] EMMS, D., WILSON, R., AND HANCOCK, E. Graph embedding using a quasi-quantum analogue of the hitting times of continuous time quantum walks. *Quantum Information & Computation 9*, 3-4 (2009), 231–254.

[53] ERDÖS, P., AND RÉNYI, A. On random graphs. *Publ. Math. Debrecen 6* (1959), 290–297.

[54] ESTRADA, E. *The structure of complex networks: theory and applications.* OUP Oxford, 2011.

[55] FARHI, E., AND GUTMANN, S. Quantum computation and decision trees. *Physical Review A 58*, 2 (1998), 915.

[56] FERRER, M., VALVENY, E., SERRATOSA, F., RIESEN, K., AND BUNKE, H. Generalized median graph computation by means of graph embedding in vector spaces. *Pattern Recognition 43*, 4 (2010), 1642–1655.

[57] FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry* (1977), 35–41.

[58] FREEMAN, L. C. Centrality in social networks conceptual clarification. *Social networks 1*, 3 (1979), 215–239.

[59] FRIEDMAN, N., AND KOLLER, D. Being bayesian about network structure. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence* (2000), Morgan Kaufmann Publishers Inc., pp. 201–210.

[60] GAERTNER, T., FLACH, P., AND WROBEL, S. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop* (August 2003), Springer-Verlag, pp. 129–143.

[61] GAO, X., XIAO, B., TAO, D., AND LI, X. A survey of graph edit distance. *Pattern Analysis and applications 13*, 1 (2010), 113–129.

[62] GODSIL, C. Average mixing of continuous quantum walks. *Journal of Combinatorial Theory, Series A 120*, 7 (2013), 1649–1662.

[63] GOLLAND, P., ERIC, W., AND GRIMSON, L. Fixed topology skeletons. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on* (2000), vol. 1, IEEE, pp. 10–17.

[64] HAMMERSLEY, J. M., HANDSCOMB, D. C., AND WEISS, G. Monte carlo methods. *Physics Today 18* (1965), 55.

[65] HAN, L., ROSSI, L., TORSELLO, A., WILSON, R., AND HANCOCK, E. Information theoretic prototype selection for unattributed graphs. *Structural, Syntactic, and Statistical Pattern Recognition* (2012), 33–41.

[66] HAN, L., WILSON, R. C., AND HANCOCK, E. R. A supergraph-based generative model. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (2010), IEEE, pp. 1566–1569.

[67] HAUSSLER, D. Convolution kernels on discrete structures. Tech. rep., Technical report, UC Santa Cruz, 1999.

[68] HE, X. C., AND YUNG, N. H. Curvature scale space corner detector with adaptive threshold and dynamic region of support. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (2004), vol. 2, IEEE, pp. 791–794.

[69] HELLINGER, E. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik 136* (1909), 210–271.

[70] HIDOVIĆ, D., AND PELILLO, M. Metrics for attributed graphs based on the maximal similarity common subgraph. *International Journal of Pattern Recognition and Artificial Intelligence 18*, 03 (2004), 299–313.

[71] HISADA, M., BELYAEV, A. G., AND KUNII, T. L. A 3d voronoi-based skeleton and associated surface features. In *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on* (2001), IEEE, pp. 89–96.

[72] ITO, T., CHIBA, T., OZAWA, R., YOSHIDA, M., HATTORI, M., AND SAKAKI, Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences 98*, 8 (2001), 4569.

[73] JEONG, H., TOMBOR, B., ALBERT, R., OLTVAI, Z., AND BARABÁSI, A. The large-scale organization of metabolic networks. *Nature 407*, 6804 (2000), 651–654.

[74] JIANG, X., MUNGER, A., AND BUNKE, H. An median graphs: properties, algorithms, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 10 (2001), 1144–1151.

[75] JOST, J. *Riemannian geometry and geometric analysis.* Springer, 2011.

[76] JU, T., SCHAEFER, S., AND WARREN, J. Mean value coordinates for closed triangular meshes. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 561–566.

[77] KALAPALA, V., SANWALANI, V., AND MOORE, C. The structure of the united states road network. *Preprint, University of New Mexico* (2003).

[78] KASS, M., WITKIN, A., AND TERZOPOULOS, D. Snakes: Active contour models. *International journal of computer vision 1*, 4 (1988), 321–331.

[79] KEMPE, J. Quantum random walks: an introductory overview. *Contemporary Physics 44*, 4 (2003), 307–327.

[80] KENDALL, D. G. Abundance matrices and seriation in archaeology. *Probability Theory and Related Fields 17*, 2 (1971), 104–112.

[81] KENDON, V. Decoherence in quantum walks-a review. *Mathematical Structures in Computer Science 17*, 6 (2007), 1169–1220.

[82] KIMMEL, R., SHAKED, D., KIRYATI, N., AND BRUCKSTEIN, A. M. Skeletonization via distance maps and level sets. In *Photonics for Industrial Applications* (1995), International Society for Optics and Photonics, pp. 137–148.

[83] KONDOR, R. I., AND LAFFERTY, J. Diffusion kernels on graphs and other discrete input spaces. In *ICML* (2002), vol. 2, pp. 315–322.

[84] KROVI, H., AND BRUN, T. Quantum walks with infinite hitting times. *Physical Review A 74*, 4 (2006), 042334.

[85] KULLBACK, S. *Information theory and statistics.* Dover publications, 1997.

[86] LAMBERTI, P., MAJTEY, A., BORRAS, A., CASAS, M., AND PLASTINO, A. Metric character of the quantum Jensen-Shannon divergence. *Physical Review A 77*, 5 (2008), 052311.

[87] LEYMARIE, F., AND LEVINE, M. D. Simulating the grassfire transform using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14*, 1 (1992), 56–75.

[88] LIN, J. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on 37*, 1 (1991), 145–151.

[89] LINDBLAD, G. Entropy, information and quantum measurements. *Communications in Mathematical Physics 33*, 4 (1973), 305–322.

[90] LODHI, H., SAUNDERS, C., SHAWE-TAYLOR, J., CRISTIANINI, N., AND WATKINS, C. Text classification using string kernels. *The Journal of Machine Learning Research 2* (2002), 419–444.

[91] MACARTHUR, B., SÁNCHEZ-GARCÍA, R., AND ANDERSON, J. Symmetry in complex networks. *Discrete Applied Mathematics 156*, 18 (2008), 3525–3531.

[92] Mahdian, M., and Xu, Y. Stochastic kronecker graphs. *Algorithms and models for the web-graph* (2007), 179–186.

[93] Majtey, A., Lamberti, P., Martin, M., and Plastino, A. Wootters' distance revisited: a new distinguishability criterium. *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics 32,* 3 (2005), 413–419.

[94] Majtey, A., Lamberti, P., and Prato, D. Jensen-Shannon divergence as a measure of distinguishability between mixed quantum states. *Physical Review A 72,* 5 (2005), 052310.

[95] Malandain, G., Bertrand, G., and Ayache, N. Topological segmentation of discrete surfaces. *International Journal of Computer Vision 10,* 2 (1993), 183–197.

[96] Martins, A. F., Smith, N. A., Xing, E. P., Aguiar, P. M., and Figueiredo, M. A. Nonextensive information theoretic kernels on measures. *The Journal of Machine Learning Research 10* (2009), 935–975.

[97] Meijster, A., Roerdink, J. B., and Hesselink, W. H. A general algorithm for computing distance transforms in linear time. In *Mathematical Morphology and its applications to image and signal processing.* Springer, 2002, pp. 331–340.

[98] Mowshowitz, A. Entropy and the complexity of graphs: I. an index of the relative complexity of a graph. *Bulletin of Mathematical Biology 30,* 1 (1968), 175–204.

[99] Mülken, O., and Blumen, A. Continuous-time quantum walks: Models for coherent transport on complex networks. *Physics Reports 502,* 2 (2011), 37–87.

[100] Nayar, S., Nene, S., and Murase, H. Columbia object image library (coil 100). Tech. rep., Tech. Report No. CUCS-006-96. Department of Comp. Science, Columbia University, 1996.

[101] Nene, S. A., Nayar, S. K., and Murase, H. Columbia object image library (coil-20). *Dept. Comput. Sci., Columbia Univ., New York.[Online] http://www. cs. columbia. edu/CAVE/coil-20. html 62* (1996).

[102] Neumann, L., Csébfalvi, B., König, A., and Gröller, E. Gradient estimation in volume data using 4d linear regression. In *Computer Graphics Forum* (2000), vol. 19, Wiley Online Library, pp. 351–358.

[103] Newman, M. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E 64,* 1 (2001), 016131.

[104] Newman, M. E. A measure of betweenness centrality based on random walks. *Social networks 27,* 1 (2005), 39–54.

[105] Nielsen, M., and Chuang, I. *Quantum computation and quantum information.* Cambridge university press, 2010.

[106] Ogniewicz, R. L., and Kübler, O. Hierarchic voronoi skeletons. *Pattern recognition 28*, 3 (1995), 343–359.

[107] Padgett, J. F., and Ansell, C. K. Robust action and the rise of the medici, 1400-1434. *American journal of sociology* (1993), 1259–1319.

[108] Passerini, F., and Severini, S. The von neumann entropy of networks. *arXiv preprint arXiv:0812.2597* (2008).

[109] Pelillo, M. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation 11*, 8 (1999), 1933–1955.

[110] Pelillo, M., Siddiqi, K., and Zucker, S. W. Matching hierarchical structures using association graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 21*, 11 (1999), 1105–1120.

[111] Quadros, W., Shimada, K., and Owen, S. 3d discrete skeleton generation by wave propagation on pr-octree for finite element mesh sizing. In *Proceedings of the ninth ACM symposium on Solid modeling and applications* (2004), Eurographics Association, pp. 327–332.

[112] Rabbat, M. G., Figueiredo, M. A., and Nowak, R. D. Network inference from co-occurrences. *Information Theory, IEEE Transactions on 54*, 9 (2008), 4053–4068.

[113] Ren, P., Wilson, R. C., and Hancock, E. R. Graph characterization via ihara coefficients. *Neural Networks, IEEE Transactions on 22*, 2 (2011), 233–245.

[114] Reniers, D., and Telea, A. Skeleton-based hierarchical shape segmentation. In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on* (2007), IEEE, pp. 179–188.

[115] Reniers, D., van Wijk, J. J., and Telea, A. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *Visualization and Computer Graphics, IEEE Transactions on 14*, 2 (2008), 355–368.

[116] Riesen, K., and Bunke, H. Iam graph database repository for graph based pattern recognition and machine learning. In *Structural, Syntactic, and Statistical Pattern Recognition.* Springer, 2008, pp. 287–297.

[117] Santha, M. Quantum walk based search algorithms. *Theory and Applications of Models of Computation* (2008), 31–46.

[118] SCHÖLKOPF, B., SMOLA, A., AND MÜLLER, K.-R. Kernel principal component analysis. In *Artificial Neural Networks—ICANN'97*. Springer, 1997, pp. 583–588.

[119] SCHÖLKOPF, B., AND SMOLA, A. J. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* MIT press, 2001.

[120] SCHWARZ, G. Estimating the dimension of a model. *The annals of statistics 6*, 2 (1978), 461–464.

[121] SHAPIRO, L., AND HARALICK, R. Structural descriptions and inexact matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 5 (1981), 504–519.

[122] SHENVI, N., KEMPE, J., AND WHALEY, K. Quantum random-walk search algorithm. *Physical Review A 67*, 5 (2003), 052307.

[123] SHEPARD, D. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference* (1968), ACM, pp. 517–524.

[124] SHERVASHIDZE, N., SCHWEITZER, P., VAN LEEUWEN, E. J., MEHLHORN, K., AND BORGWARDT, K. M. Weisfeiler-lehman graph kernels. *The Journal of Machine Learning Research 12* (2011), 2539–2561.

[125] SHERVASHIDZE, N., VISHWANATHAN, S., PETRI, T., MEHLHORN, K., AND BORGWARDT, K. Efficient graphlet kernels for large graph comparison. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics. Society for Artificial Intelligence and Statistics* (2009).

[126] SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. The princeton shape benchmark. In *Shape Modeling Applications, 2004. Proceedings* (2004), IEEE, pp. 167–178.

[127] SIDDIQI, K., BOUIX, S., TANNENBAUM, A., AND ZUCKER, S. W. The hamilton-jacobi skeleton. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (1999), vol. 2, IEEE, pp. 828–834.

[128] SIDDIQI, K., BOUIX, S., TANNENBAUM, A., AND ZUCKER, S. W. Hamilton-jacobi skeletons. *International Journal of Computer Vision 48*, 3 (2002), 215–231.

[129] SIDDIQI, K., AND PIZER, S. M. *Medial representations: mathematics, algorithms and applications*, vol. 37. Springer, 2008.

[130] SINKHORN, R. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics 35*, 2 (1964), 876–879.

[131] Sporns, O. Network analysis, complexity, and brain function. *Complexity 8*, 1 (2002), 56–60.

[132] Sun, J., Ovsjanikov, M., and Guibas, L. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1383–1392.

[133] Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323.

[134] Todorovic, S., and Ahuja, N. Extracting subimages of an unknown category from a set of images. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 1, IEEE, pp. 927–934.

[135] Torsello, A. An importance sampling approach to learning structural representations of shape. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–7.

[136] Torsello, A., and Dowe, D. L. Learning a generative model for structural representations. In *AI 2008: Advances in Artificial Intelligence*. Springer, 2008, pp. 573–583.

[137] Torsello, A., and Hancock, E. R. A skeletal measure of 2d shape similarity. *Computer Vision and Image Understanding 95*, 1 (2004), 1–29.

[138] Torsello, A., and Hancock, E. R. Correcting curvature-density effects in the hamilton–jacobi skeleton. *Image Processing, IEEE Transactions on 15*, 4 (2006), 877–891.

[139] Torsello, A., and Hancock, E. R. Learning shape-classes using a mixture of tree-unions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 6 (2006), 954–967.

[140] Torsello, A., Hidovic-Rowe, D., and Pelillo, M. Polynomial-time metrics for attributed trees. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 27*, 7 (2005), 1087–1099.

[141] Vapnik, V. Statistical learning theory, 1998.

[142] Vishwanathan, S., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. Graph kernels. *The Journal of Machine Learning Research 99* (2010), 1201–1242.

[143] Wallace, C. S., and Dowe, D. L. Minimum message length and kolmogorov complexity. *The Computer Journal 42*, 4 (1999), 270–283.

[144] WATTS, D., AND STROGATZ, S. The small world problem. *Collective Dynamics of Small-World Networks 393* (1998), 440–442.

[145] WEST, G., BROWN, J., AND ENQUIST, B. A general model for the structure, function, and allometry of plant vascular systems. *Nature 400* (1999), 664–667.

[146] WHITE, D., AND WILSON, R. C. Spectral generative models for graphs. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on* (2007), IEEE, pp. 35–42.

[147] WITTEN, I. H., AND FRANK, E. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.

[148] WOOTTERS, W. Statistical distance and hilbert space. *Physical Review D 23*, 2 (1981), 357.

[149] XIAO, Y., XIONG, M., WANG, W., AND WANG, H. Emergence of symmetry in complex networks. *Phys. Rev. E 77* (Jun 2008), 066108.

[150] YOSHIZAWA, S., BELYAEV, A., AND SEIDEL, H.-P. Skeleton-based variational mesh deformations. In *Computer Graphics Forum* (2007), vol. 26, Wiley Online Library, pp. 255–264.

[151] ZACHARY, W. W. An information flow model for conflict and fission in small groups. *Journal of anthropological research* (1977), 452–473.

[152] ZHANG, J., SIDDIQI, K., MACRINI, D., SHOKOUFANDEH, A., AND DICKINSON, S. Retrieving articulated 3-d models using medial surfaces and their graph spectra. In *Energy minimization methods in computer vision and pattern recognition* (2005), Springer, pp. 285–300.