
# Shape Analysis and Description Based on the Isometric Invariances of Topological Skeletonization

# Descripción y análisis de forma basado en la invarianza a isometrias de los esqueletos topológicos

**Diego Alberto Patiño Cortés**

Universidad Nacional de Colombia - Sede Medellín
Facultad de Minas
Departamento de Ciencias de la Computación y la Decisión
Medellín, Colombia
2019

# Shape Analysis and Description Based on the Isometric Invariances of Topological Skeletonization

# Descripción y análisis de forma basado en la invarianza a isometrias de los esqueletos topológicos

**Diego Alberto Patiño Cortés**

In Partial Fulfillment of the Requirements for the Degree of:
Doctor of Engineering - Systems Engineering

**Advisor:**
John Willian Branch Bedoya, PhD,

**Research Area:**
Computer Vision and Machine Learning
**Research group:**
GIDIA - Grupo de Investigación en Inteligencia Artificial

Universidad Nacional de Colombia - Sede Medellín
Facultad de Minas
Departamento de Ciencias de la Computación y la Decisión
Medellín, Colombia
2019

**Dedication:**

I dedicate this study my parents, Nora and Carlos. They taught me the most important lessons that have made me the person I am right now, especially the importance of education.
And, to my little brother, Andrés, who I love and has always offered me his unconditional support.

# Acknowledgement

I want to thank all my extended family. My mother Nora, my father Carlos, and my little brother Andrés. All of them were essential pieces in my life during this period.
To my best friend David Saldaña, and my girlfriend Leslie. They have been with me in the roughest, intense, and most stressful parts of completing this work. And yet, they have helped, listened and tolerated me. Thank you from the bottom of my heart.
To my dear friends: Juan Carlos Rodriguez, Julian Sepúlveda, Erika Figueroa, Carlos Restrepo, Melissa Dominguez, Milton Alvardo, Daniela Zapata, Melissa Agudelo, Marian Bonfim, Carlos Esteves, Christine Allen, Andreea Alexandru, Sergio Gutierrez, Mariana Vásquez, and Pedro Atencio. Thanks for all the support during these years.
I also want to thank my two main research groups during this period: GIDIA lab at Universidad Nacional de Colombia, and GRASP LAB at the University of Pennsylvania. The meaningful discussions, the friendly environment, and the fun coffee breaks kept me motivated all this time.
Finally, I want to thank my funding agency, Departamento Administrativo de Ciencia, Tecnología e Innovación (Colciencias). They provided the resources to complete my studies.

# Abstract

In this dissertation, we explore the problem of how to describe the shape of an object in 2D and 3D with a set of features that are invariant to isometric transformations. We focus to based our approach on the well-known Medial Axis Transform and its topological properties. We aim to study two problems. The first is how to find a shape representation of a segmented object that exhibits rotation, translation, and reflection invariance. The second problem is how to build a machine learning pipeline that uses the isometric invariance of the shape representation to do both classification and retrieval. Our proposed solution demonstrates competitive results compared to state-of-the-art approaches.

We based our shape representation on the medial axis transform (MAT), sometimes called the topological skeleton. Accepted and well-studied properties of the medial axis include: homotopy preservation, rotation invariance, mediality, one pixel thickness, and the ability to fully reconstruct the object.

These properties make the MAT a suitable input to create shape features; however, several problems arise because not all skeletonization methods satisfy all the above-mentioned properties at the same time. In general, skeletons based on thinning approaches preserve topology but are noise sensitive and do not allow a proper reconstruction. They are also not invariant to rotations. Voronoi skeletons also preserve topology and are rotation invariant, but do not have information about the thickness of the object, making reconstruction impossible. The Voronoi skeleton is an approximation of the real skeleton. The denser the sampling of the boundary, the better the approximation; however, a denser sampling makes the Voronoi diagram more computationally expensive.

In contrast, distance transform methods allow the reconstruction of the original object by providing the distance from every pixel in the skeleton to the boundary. Moreover, they exhibit an acceptable degree of the properties listed above, but noise sensitivity remains an issue. Therefore, we selected distance transform medial axis methods as our skeletonization strategy, and focused on creating a new noise-free approach to solve the contour noise problem.

To effectively classify an object, or perform any other task with features based on its shape, the descriptor needs to be a normalized, compact form: $\Phi$ should map every shape $\Omega$ to the same vector space $\mathrm{R}^n$. This is not possible with skeletonization methods because the skeletons of different objects have different numbers of branches and different numbers of points, even when they belong to the same category. Consequently, we developed a strategy to extract features from the skeleton through the map $\Phi$, which we used as an input to a machine learning approach.

After developing our method for robust skeletonization, the next step is to use such skeleton into the machine learning pipeline to classify object into previously defined categories. We developed a set of skeletal features that were used as input data to the machine learning

architectures. We ran experiments on MPEG7 and ModelNet40 dataset to test our approach in both 2D and 3D. Our experiments show results comparable with the state-of-the-art in shape classification and retrieval. Our experiments also show that our pipeline and our skeletal features exhibit some degree of invariance to isometric transformations.
In this study, we sought to design an isometric invariant shape descriptor through robust skeletonization enforced by a feature extraction pipeline that exploits such invariance through a machine learning methodology. We conducted a set of classification and retrieval experiments over well-known benchmarks to validate our proposed method.

**Keywords:** Medial Axis Transform, Isometry, Morphological Skeletonization, Shape Analysis and Description, Shape feature, Invariance and Equivariance, PointNet, Chordiogam, Shape Classification and Retrieval

# Resumen

En esta disertación se explora el problema de cómo describir la forma de un objeto en 2D y 3D con un conjunto de características que sean invariantes a transformaciones isométricas. La metodología propuesta en este documento se enfoca en la Transformada del Eje Medio (Medial Axis Transform) y sus propiedades topológicas. Nuestro objetivo es estudiar dos problemas. El primero es encontrar una representación matemática de la forma de un objeto que exhiba invarianza a las operaciones de rotación, translación y reflexión. El segundo problema es como construir un modelo de machine learning que use esas invarianzas para las tareas de clasificación y consulta de objetos a través de su forma. El método propuesto en esta tesis muestra resultados competitivos en comparación con otros métodos del estado del arte.

En este trabajo basamos nuestra representación de forma en la transformada del eje medio, a veces llamada esqueleto topológico. Algunas propiedades conocidas y bien estudiadas de la transformada del eje medio son: conservación de la homotopía, invarianza a la rotación, su grosor consiste en un solo pixel (1D), y la habilidad para reconstruir el objeto original a través de ella.

Estas propiedades hacen de la transformada del eje medio un punto de partida adecuado para crear características de forma. Sin embargo, en este punto surgen varios problemas dado que no todos los métodos de esqueletización satisfacen, al mismo tiempo, todas las propiedades mencionadas anteriormente. En general, los esqueletos basados en enfoques de erosión morfológica conservan la topología del objeto, pero son sensibles al ruido y no permiten una reconstrucción adecuada. Además, no son invariantes a las rotaciones. Otro método de esqueletización son los esqueletos de Voronoi. Los esqueletos de Voronoi también conservan la topología y son invariantes a la rotación, pero no tienen información sobre el grosor del objeto, lo que hace imposible su reconstrucción. Cuanto más denso sea el muestreo del contorno del objeto, mejor será la aproximación. Sin embargo, un muestreo más denso hace que el diagrama de Voronoi sea más costoso computacionalmente.

Por el contrario, los métodos basados en la transformada de la distancia permiten la reconstrucción del objeto original, ya que proporcionan la distancia desde cada píxel del esqueleto hasta su punto más cercano en el contorno. Además, exhiben un grado aceptable de las propiedades enumeradas anteriormente, aunque la sensibilidad al ruido sigue siendo un problema. Por lo tanto, en este documento seleccionamos los métodos basados en la transformada de la distancia como nuestra estrategia de esqueletización, y nos enfocamos en crear un nuevo enfoque que resuelva el problema del ruido en el contorno.

Para clasificar eficazmente un objeto o realizar cualquier otra tarea con características basadas en su forma, el descriptor debe ser compacto y estar normalizado: $\Phi$ debe relacionar cada forma $\Omega$ al mismo espacio vectorial $\mathrm{R}^n$. Esto no es posible con los métodos de esqueletización en el estado del arte, porque los esqueletos de diferentes objetos tienen

diferentes números de ramas y diferentes números de puntos incluso cuando pertenecen a la misma categoría. Consecuentemente, en nuestra propuesta desarrollamos una estrategia para extraer características del esqueleto a través de la función $\Phi$, que usamos como entrada para un enfoque de aprendizaje automático.
Después de desarrollar nuestro método de esqueletización robusta, el siguiente paso es usar dicho esqueleto en un modelo de aprendizaje de máquina para clasificar el objeto en categorías previamente definidas. Para ello se desarrolló un conjunto de características basadas en el eje medio que se utilizaron como datos de entrada para la arquitectura de aprendizaje automático.
Realizamos experimentos en los conjuntos de datos: MPEG7 y ModelNet40 para probar nuestro enfoque tanto en 2D como en 3D. Nuestros experimentos muestran resultados comparables con el estado del arte en clasificación y consulta de formas (retrieval). Nuestros experimentos también muestran que el modelo desarrollado junto con nuestras características basadas en el eje medio son invariantes a las transformaciones isométricas.

**Keywords:** Transformada del eje medio, Isometría, Esqueletos topológicos, Análisis y descripción de forma, Invarianza y equivarianza, PointNet, Chordiogam, Clasificación y recuperación de formas

# Contents

# List of Figures

# List of Tables

# List of Symbols and Abbreviations

| Abbreviation | Description |
|---|---|
| MAT | Medial Axis Transform |
| CNN | Convolutional Neural Network |
| CBD | Contour based descriptors |
| FD | Fourie Descriptor |
| RBD | Region-based methods |
| ART | The Angular Radial Transform |
| $\Delta$, $\nabla^2$ | Laplacian operator |
| SC | Shape Context |
| $\Omega$ | A shape region in $\mathbb{R}^n$ |
| $\delta\Omega$ | Boundary of a shape $\Omega$ |
| MVCNN | Multiview CNNs |
| SO(3) | The 3D rotation group |
| $d_H$ | Hausdorff distance |
| $d_D$ | Dubuisson-Jain dissimilarity |
| DCT | Discrete Cosine Transform |
| SAT | Scale Axis Transform |
| RMA | Real Medial Axis |
| GIMA | Gamma($\gamma$) Integer Medial Axis |
| CPMA | Cosine-Pruned Medial Axis |
| SBOR | Shape-Based Object Retrieval |
| mAP | Mean Average Precision |
| SVM | Support Vector Machine |
| VD | Voronoi Diagram |
| MLP | Multilayer Perceptron |

# 1 Introduction

Understanding the shape of objects in the real world has been a critically important issue for centuries, even millennia. Many great minds in the history of the world have dedicated themselves to unraveling the meaning of the concept of shape. The results of this exhaustive exploration have expanded our understanding of shape from early concepts as the Platonic solids, to more complex concepts encompassed by theories such as differential geometry (do Carmo, 1992), the formulation of Gestalt psychology (Koffka, 1999), Kendall's shape analysis (Kendall, 1984a), and more.

Shapes of objects arise naturally in many fields where the geometric information of volumes or surfaces plays an essential role in the subject of study. A clear example of this phenomenon is the field medicine, where many clinical applications such as radiotherapy planning, MRI analysis, image-guided surgery, and treatment evolution (Nava-Yazdani et al., 2019) heavily rely on the analysis and processing of both 2D and 3D data. There are many additional areas in which understanding shapes can be a useful tool. Examples of such areas include non-destructive object study and reconstruction in archaeology and cultural heritage (Tal, 2014; van der Maaten et al., 2006); object classification and retrieval from large collection of images (Li et al., 2018; Safar and Shahabi, 2003); human action and pose recognition for gaming and entertainment (Chaudhry et al., 2013; Li et al., 2019); environment sensing in robot navigation and planning (Peters and Ledoux, 2016; Li et al., 2017); and industry for automatic visual quality inspection of product defects (Qiu et al., 2011).

Computational-driven study of shape has been a matter of interest since the invention of modern computers through fields like computer vision, computational geometry, and machine learning. Therefore, it is unsurprising that technological advances are improving the way we store shapes in digital formats, e.g. digital images (2D), point clouds and triangular meshes (3D).

The analysis of 3D models is very different from the analysis of 2D images. The main difference is that 3D models offer a complete representation of the object since occlusions are more likely to be present in projected data in 2D. However, 3D processing and analysis are much more complicated to handle and model compared to 2D pixels in a regular grid. Hence, 2D descriptors usually do not generalize to 3D, splitting classification and retrieval pipelines in two.

Shape analysis relies on an accurate mathematical representation of the shape of an object, regardless of the domain of applications. This representation should be able to exhibit geometric and topological properties inherent to the shape itself. As a result, we can define

shape analysis as *"a set of theories, methods, and algorithms that concur to the formalization and computation of properties useful to characterize the geometrical appearance of objects."* (Biasotti et al., 2014).
There are two key aspects to shape analysis. The first key aspect it the object presentation that determines how the geometric information of the object is stored. Examples of the most common representations are point clouds, contour curves, meshes, and topological skeletons. The latter being one of the focuses of our work in this study. The second aspect centers on how to extract features from a chosen representation. Such features should highlight the geometric properties of the shape. Both representation and description are essential for shape analysis.

## 1.1 Problem statement

Researchers have proposed a variety of shape descriptors. These descriptors differ widely in their mathematical formulation. They are designed to address specific goals in a particular field. The purpose of every shape descriptor is to provide a compact but complete set of features of the shape of an object, preserving its appearance and geometric properties. For a 3D model compact representation is preferred because the amount of data required to represent the entire structure of an object is exponentially higher than for 2D models.
Over the past two decades, computer vision research has put enormous effort into creating more effective and efficient shape descriptors. Although we have witnessed significant progress, especially in the last ten years, performance remains unsatisfactory. There are several challenges to shape description that guide ongoing research in the field of computer vision.
One of these challenges is isometric invariance and equivariance, e.g., a shape descriptor should be rotation invariant. In other words, a feature map $f$ should describe an object in the same way, regardless of its pose. Rotation, like any other isometric transformation, preserves the geometric properties of the object. Many shape descriptors in literature are not rotation invariant by design, despite their good performance in tasks such as shape classification and retrieval (Worrall and Brostow, 2018; Kondor and Trivedi, 2018). A clear example of this is approaches based on CNNs, where the base operation is the mathematical the convolution, which is equivariant to translations but not to rotations. A particularly difficult result of this non-invariance problem occurs when dealing with arbitrary rotations in $\mathbf{SO}(3)$, the mathematical group describing all possible rotations in a 3D vector space. Some authors approach this problem by setting a canonical pose for the objects. However, there is no explicit agreement about what a canonical pose means.
Additionally, many shape description methods also encounter difficulties when dealing with shapes of the same category but at different scales. This scenario is common due to the wide variety of sensors. e.g., cameras, point clouds, scanners, etc. Authors use object normalization to tackle this problem, enclosing the object in a normalized boundary. However, in more

extreme settings such as images in the wild, clutter, or complex background, normalization is challenging to conduct.

Intra-class variation is also a common problem. It occurs when objects belong to a single category but have significant differences with that category, e.g., chairs with and without arms, with wheeled legs, etc. Another challenge related to intra-class variation is the difference in topology (Genus number or number of holes) among different objects, even when they belong to the same class. It is common for a pair of objects belonging to the same class to differ in the number of holes in their structure. These differences make it difficult for many descriptors to produce a distinctive set of features for the specific class, particularly when the descriptors do not know the specific topology of the objects beforehand.

Finally, another frequent problem is shape non-rigid deformation. Objects in categories such as cats, horses, or human beings, are susceptible to having deformations due to the movement of their bodies. While these deformations do not change the category the shape belongs to, they do make it more difficult to describe the object accurately in every possible pose.

Graph-based methods, especially those based on the Medial Axis Transform (also called topological skeleton), have interesting properties that can address the shape description challenges detailed above. They provide dimensionality reduction while preserving the topology. They are rotation invariant because the medial axis of a rotated object is the rotated medial axis of the original one. The MAT is also robust to small deformation, such as articulation because of its graph-like structure. For example, the movement of a human-like shape does not affect the entire skeleton but only the connections between the nodes of the skeleton.

However, there are also downsides to using this methodology. Skeletons are extremely sensitive to the noise of the contour or surfaces of the object. Even small amounts of noise can cause erroneous sections of the skeleton to appear. Moreover, skeletons of some shapes are ambiguous, meaning that two somewhat different shapes can have similar skeletal representations. Therefore, skeletonization methods are usually used in combination with other approaches.

In this dissertation, we studied shape analysis for classification and retrieval. We focused our work on designing a new shape description strategy with invariant properties to isometric transformation. We used the Medial Axis Transform as our shape representation because of its properties that make it invariant to isometries. Due to its extreme sensitivity to noise, we first formulated a robust skeletonization algorithm capable of estimating the "true skeleton" of an object with fewer spurious branches. We designed a machine learning approach to extract shape features from the medial axis transform, that can be applied to 2D as well as 3D shapes. We conducted shape classification and retrieval experiments in order to assess the advantages of our approach against state-of-the-art methods. Figure **1-1** shows an overview of our methodology.

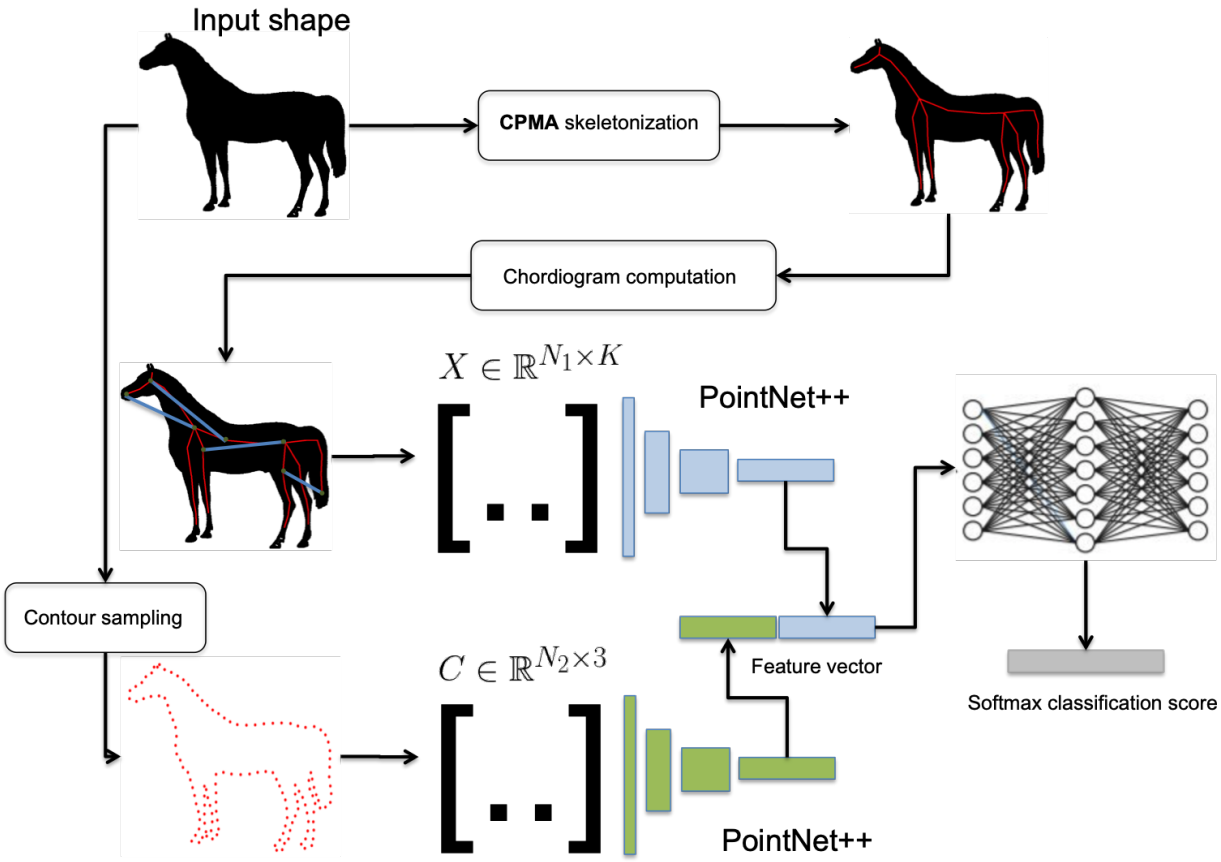



**Figure 1-1:** Proposed Machine learning pipeline. The CPMA skeleton of the input shape is computed to extract chord (blue lines) later. Both the chords and a sampled point cloud from the boundary are sent to independent PointNet++ architectures, and later passed through a fully connected network to get the classification scores.

## 1.2 Objectives

In this dissertation, we study shape analysis with the ultimate goal of defining a shape descriptor capable of exhibiting isometric invariance for shape classification and retrieval. We explore shape description methods; specifically, those that depend on the Medial Axis and its topology-preserving properties. With this in mind, we define the following objectives for this study:

### English version

**General objective:** Study how to formulate a shape description of an object in a digital image through its Medial Axis. The feature representation focused on the invariance to isometries that are inherent to the Medial Axis.

#### Detailed objectives

1. Analyze and categorize the main Medial Axis-Based shape descriptors that exist in the literature, listing their properties and assessing their advantages and weaknesses.

2. Identify the most relevant 2D and 3D skeletonization methods in the state-of-the-art that are robust to boundary/surface noise of the segmented object.

3. Design a new Medial Axis-based shape descriptor that works in 2D and 3D, which exhibits invariance to isometries. Without being the main focus of our study, scale and small non-rigid deformations invariance are also considered.

4. Test the stated shape descriptor through *Shape Classification* and *Shape Retrieval* experiments, over relevant benchmarks and dataset in the field.

### Spanish version

**Objetivo general** Estudiar la descripción de la forma de un objeto en una imagen digital a partir de su representación como un esqueleto topológico. El estudio está enfocado en sus las propiedades invariantes a isometrías, escala y pequeñas deformaciones no rígidas.

#### Objetivos específicos

1. Analizar los principales métodos existentes para representar la forma de un objeto en una imagen digital a partir de su esqueleto topológico y enumerar sus principales ventajas y desventajas en torno a sus propiedades.

2. Identificar en la literatura existente un método de esqueletonización robusto en 2D y 3D, que muestre poca sensibilidad al ruido presente en la segmentación del objeto.

3. Proponer un nuevo descriptor de forma en 2D y 3D basado en el concepto de esqueleto topologico (Medial Axis); con propiedades invariantes a isometrías, escala y pequeñas deformaciones no rígidas.

4. Evaluar el desempeño del descriptor propuesto comparándolo con los principales benchmarks utilizados en la comunidad científica a través de experimentos de *shape classification* y *shape retrieval*. La evaluación resaltará las propiedades, ventajas y desventajas del descriptor con respecto a otros enfoques.

## 1.3 Contributions

Shape Description has proliferated in the field of computer vision. We focused on the topological properties of shapes in 2D and 3D through the study of the concept of the Medial

Axis Transform. We designed a shape descriptor based on the MAT by exploiting its invariance to topology, scale, isometries, and small deformations. We tested our methodology by conducting several classification experiments on state-of-the-art datasets.
Additionally, several academic papers were submitted to conferences and peer-reviewed journals in the field of computer vision. These papers summarize the main contribution of this dissertation:

1. In chapter 4 and 5, we explore robust skeletonization methods. Robustness, in this case, is measured as the degree of invariance to noise and isometric transformation. We were able to identify, use, and evaluate the main approaches in the literature. Moreover, two new methods for skeletonization were developed, compared with the state-of-the-art, and submitted to the scientific community:

   D. Patino, and J. W. Branch. "Noise-invariant skeletonization by modeling contour and surface noise in 2D and 3D objects". (Submitted to *Revista Dyna.*)

   D. Patino, and J. W. Branch. "Cosine-Pruned Medial Axis: A new strategy for spurious branch removal in 2D and 3D by constraining the cosine transform reconstruction". (Submitted to *IET Computer Vision*)

2. In chapter 6 we crafted a new classification method to exploit and demonstrate the properties of skeleton-based shape description. We tested our approach through classification and retrieval experiments.

   D. Patino, and J. W. Branch. "2D and 3D Shape classification and retrieval using noise-free topological skeletons". *Paper on progress.*

3. In addition to the listed papers, we submitted another two indirect contributions resulting from our exploration of how superpixels can be used to approximate/segment the shape of an object. In this paper, we use superpixels to segment skin lesions in dermoscopic images.

   D. Patiño, J. Avendaño, and J. W. Branch. "Automatic skin lesion segmentation on dermoscopic images by the means of superpixel merging". *International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer. 2018. p728-736.*

   Diego Patiño, Alberto Ceballos, Jairo Rodriguez, German Sánchez, and John Branch. "Melanoma Detection on Dermoscopic Images using Superpixels segmentation and Shape Based Features". *Evento de Investigacción y Socialización de Ciencias de la Computación (SICC), Medellín, Colombia.*

## 1.4 Organization

In the next chapters, we will further approach in detail the stated research topic. The remainder of this document is organized as follows. In chapter 2, we describe the preliminary

concepts that help to understand this study. In chapter 3, we offer a comprehensive literature review of shape analysis and shape description. The detailed methodology employed in the development of this dissertation is explained in chapter 4. We present and discuss our results in two chapters. First, in chapter 5, we formally define our skeletonization strategy and report the results of the robust skeletonization experiments. Later, in chapter 6, we fully describe the development of a shape classification method based on the Medial Axis Transform, providing experiments to show its advantages over other state-of-the-art approaches. Finally, in chapter 7, we draw some conclusions and present a short discussion about the study, along with future directions for this research.

# 2 Shape Analysis

The questions: what is "Shape" and how do we understand it? Has captivated mathematicians and philosophers for over a century, as seen in essential studies such as Gestalt Physiology (KING et al., 1994), Thompson's "On Growth and Form" (Thompson, 1942), or Kendall's Statistical Theory of Shape Kendall (1984b); Stoyan (1989). With the rise of Computer Science and the advent of the digital imagery era, shape analysis has become a significant area of research, penetrating application domains like medical imaging, industrial design, entertainment, computational anatomy, sensor measurement, and geographical profiling, among others. Today, shape analysis plays an integral role in the fields of computer vision and augmented reality due to their need for more diverse and precise tools to analyze vast pools of 2D and 3D data.

While the history of shape analysis is long, the area remains open to new and exciting innovation, as research problems such as shape matching, shape retrieval, classification, and semantic segmentation remain unsolved. A significant amount of research on the subject is still active, with new theories and technological advances taking place all the time. The goal of this chapter is to present the key most important concepts related to shape analysis. These concepts will be described in the following sections to enable readers to better understand the research conducted in this dissertation.

## 2.1 Shape Description

Kendall Shape Theory (Kendall, 1977) defines a shape as "all the geometrical information that remains when location, scale, and rotational effects (Euclidean transformations) are filtered out from an object." In essence, this definition implies that shape does not change when we apply any of the above mention transformations. However, one might argue that an object maintains the category it belongs to even after a non-rigid deformation. It is this type of ambiguities the ones that make shape description a challenging problem. See Figure **2-1**.

Thus, shape analysis can be defined as *the automatic analysis of geometric shapes*, for example, using a computer to detect similarly shaped objects in a database or parts that fit together. In order to perform such analysis, it is crucial to be able to describe a shape as a list of features of a vector space $f \in V$, such that $f$ summarizes the most important properties of the object. Generally speaking, a shape descriptor is a simplified representation of a

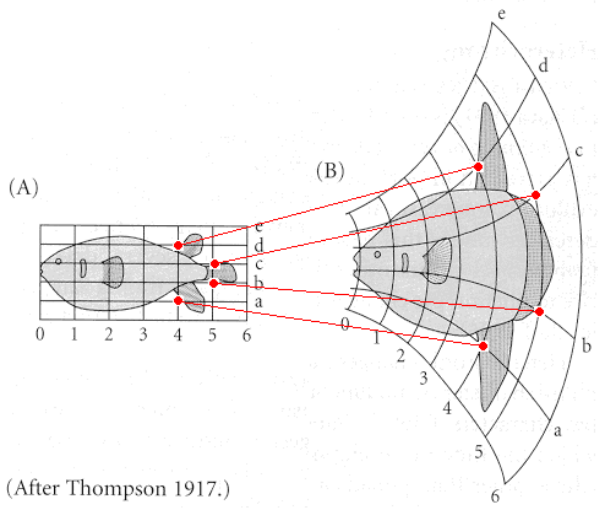



**Figure 2-1:** Non-rigid transformation between similar biological shapes. Both instances belong to the same class - fish - however, its morphology is different. On Growth and Form, D. W. Thomposon, 1997

2D or 3D shape in the form of a vector containing a set of numerical values or a graph-like structure used to describe the shape geometrically or topologically.
Here it is important to distinguish between *shape description* and *shape representation*. A short but meaningful distinction between these two concepts is offered by (Lee, 1984) as follows: "An object representation contains enough information to reconstruct (an approximation to) the object, while a description only contains enough information to identify an object as a member of some class." In other words, the representation of an object is more detailed and accurate than the description. The description is more concise and designed to highlight the particular properties of the shape. In this sense, we can think of a shape representation as an alternative way to store all of the shape's information in a different format that particularly benefits: speed, compactness and efficiency (Siddiqi et al., 1999; Toshev et al., 2012; Marie et al., 2016; Freifeld and Black, 2012).

## 2.2 Shape Matching

Shape matching is the process of comparing a pair of geometric objects by applying the notion of shape similarity. The goal of shape matching is to estimate how close or different these two shapes are. This is done to establish a set of correspondences between sections of different shapes or to find a shape similar to a model in a cluttered image/3D model. A distance applied to the features in the shape description of each object is the standard similarity measure used to make the comparison (See Figure **2-2**). The most common metrics employed for shape matching are the Hausdorff Distance, the Chamfer Distance (Butt and Maragos, 1998), or any geodesic metric in general (Ling et al., 2007).

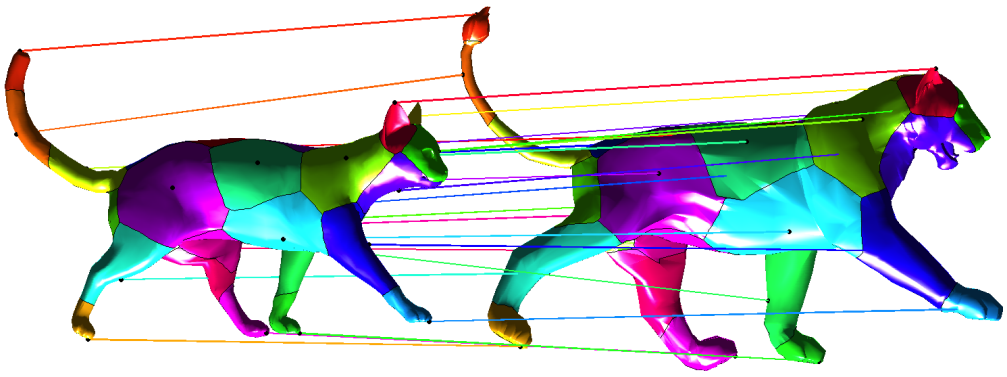


**Figure 2-2:** Shape matching example. Are the two shape similar enough to be considered in the same class?. Taken from: Computer Vision Group, TUM Department of Informatics, Technical University of Munich

Shape matching finds applications in many fields such as shape retrieval, recognizing object categories, fingerprint identification, Optical Character Recognition (OCR), and Molecular Biology.
To achieve significant results when comparing objects, we need Any shape matching approach to use metrics and a shape descriptors with some degree or equivariance or invariance to intrinsic variations between the compared shapes, i.e., rotations, scale, appearance.

## 2.3 Shape Classification and Retrieval

Shape classification in computer vision addresses the problem of assigning labels to objects belonging to one of many classes based on the object's shape. This problem is closely related to shape-based object retrieval, for which the goal is to return objects from a database that are the most similar to a predefined query object. Shape classification performance is commonly evaluated through classification accuracy per instance—more objects classified in the correct class increase the accuracy to a maximum of 100%. For shape retrieval, the preferred evaluation metric is the Mean Average Precision (mAP) over a set of queries on a dataset.
Classification is usually conducted by training machine learning algorithms with a set of hand-labeled objects. The classification algorithm learns how to divide the feature space of the shape description such that objects belonging to the same classes will cluster together. The shape description of each object is crucial for classification because it emphasizes the characteristics of the different classes.
This same description is also used for shape retrieval. In shape retrieval, we compare two objects of the same database through a positive score value $s > 0$ computed with any distance metric: $s = d(\Omega_1, \Omega_2)$. Higher values of $s$ indicate more difference between the two objects, while lower values suggest a greater similarity between them. Figure **2-3** illustrates both

shape classification and retrieval.

Although extensive research has been done on this field, shape classification is still considered as an open problem. This is, in part, attributed to factors like the large intra-class variation between shapes of the same category. It is also attributed to the wide variety of representation formats: contour data, point clouds, polygonal meshes, graphs, and signed distance functions.

Additional challenges in shape classification and retrieval include variations of the objects due to pose, deformation, occlusion, or cluttered and complex topologies.

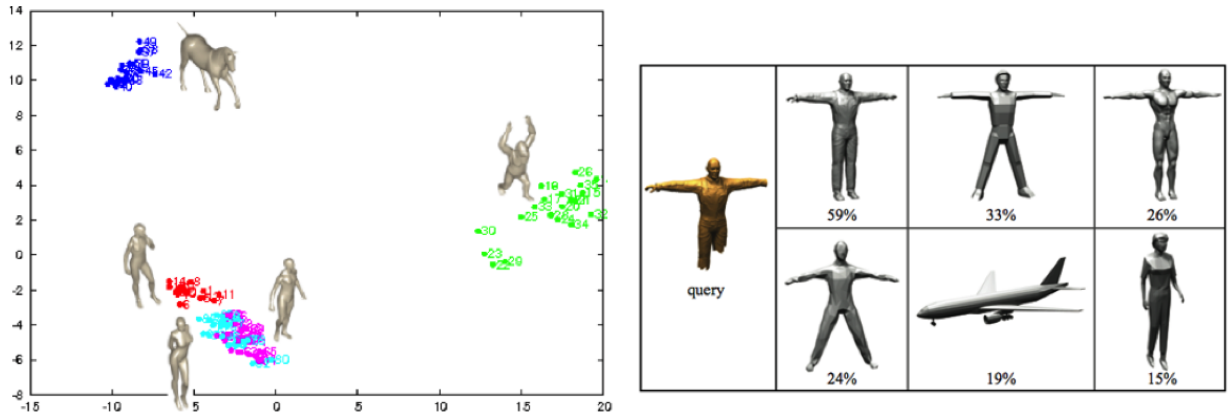



**Figure 2-3:** Shape Classification and Retrieval. Object belonging to different classes appear distant in the features space (left). Objects with similar types to the query object have higher retrieval scores than compared to distance ones (right).

## 2.4 Invariance and Equivariance

A map $f$ is said to be an equivariant map when its domain and co-domain are acted on by the same symmetry group, and the function commutes with the action of the group. This means that applying a symmetry transformation and then computing the function produces the same result as computing the function and then applying the transformation:

$$f(g.x) = g.f(x), \tag{2-1}$$

Where $g \in G$, is an element of some mathematical symmetry group. The map $f$ is called an *Invariant map*, if applying $f$ to $x$ is the same as applying $f$ to $g.x$:

$$f(g.x) = f(x), \tag{2-2}$$

In equation 2-1 and 2-2, the map $f$ might take the form of an analytically expression as well as a CNN, a SVM, etc.

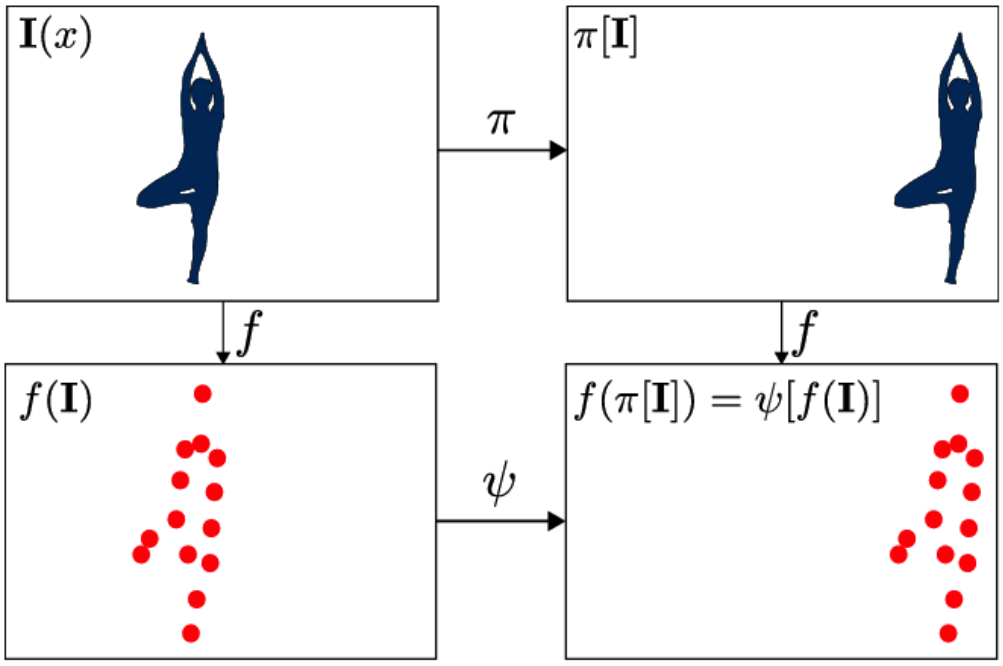



**Figure 2-4:** Equivariance example. The resulting point cloud is the same after applying the map $f$ followed by the transformation $\pi$ or viceversa (Worrall et al., 2017)

.

In the geometry of triangles, the area and perimeter of a triangle are invariant properties. Translating or rotating a triangle does not change its area or perimeter. However, triangle centers such as the centroid, circumcenter, incenter, and orthocenter are not invariant, because moving a triangle will cause its centers to move. Instead, these centers are equivariant. Applying any Euclidean congruence (a combination of a translation and rotation) to a triangle, and then constructing its center, produces the same point as constructing the center first, and then applying the same congruence to the center. See Figure **2-4**.
Informally, invariant maps demonstrate that a shape descriptor does not change with the transformation of the object. Therefore, equivariance is generally preferable because the mathematical process that makes the descriptor changes according to the transformation is known. Equivariance allows one to recover the transformation itself by mapping two objects that are known to be the same, but in different transformation/poses.

## 2.5 Isometric transformations

Isometries are geometric transformations that preserve measures of length, area and angles. These transformations include reflections, translations, and rotations. Two isometric trans-

formation $T_a$ and $T_b$ can be combined to produce a new one: $T_c = T_a T_b$. This property is in general not commutative. Formally an isometry is defined as follows:

**Definition 1** ***Isometric transformation*** *Let $X$ and $Y$ be metric spaces with metrics $d_X$ and $d_Y$. A map $f : X \mapsto Y$ is called an isometry or distance preserving if for any $a, b \in X$ one has*

$$d_Y\left(f(a), f(b)\right) = d_X(a, b) \tag{2-3}$$

According to the above definition, other geometric transformation such as projections, shearing, and scales; can not be considered isometries.
An isometry is automatically injective. If they were not, two distinct points, $a$ and $b$, could be mapped to the same point, thereby contradicting the coincidence axiom of the metric $d$.
A global isometry, isometric isomorphism, or congruence mapping is a bijective isometry. Like any other bijection, a global isometry has a function inverse. The inverse of a global isometry is also a global isometry.
Isometric invariance and equivariance in shape analysis is an important research area in computer vision because shapes naturally manifest rich variability among instances of the same class. All the geometric properties that make the object belong to the class in question remain after an isometric transformation. Any kind of shape analysis is often required to be invariant to isometric transformations and shape variations such as different poses, rotations or different canonical coordinate systems of the data, and translation of the object.
Equivariance to other transformation such scale or non-rigid deformations are also an active focus of research. These type of transformation are important because they commonly occur in articulated shapes, or due to different registration sensors.

## 2.6 Digital Shape Formats

In this section, we describe some of the most common ways to represent the shape of an object in digital formats.

**Pixels and Voxels** in digital imaging, a pixel is a physical point in a raster image. It is the smallest controllable element of a picture represented on the screen. For storing a shape, pixels take binary values $p \in [0, 1]$ representing binary occupancy. A key notion when working with pixels is the neighborhood, $N_p = \{q \in \mathbb{Z} | d(p, q) \leq \epsilon\}$. This is a set of all of the points that are immediately close to a point $p$. A shape is composed of all of the pixels with value 1 defining the area or volume of the object. The analogous of pixels in 3D are voxels that satisfy the same properties described above.

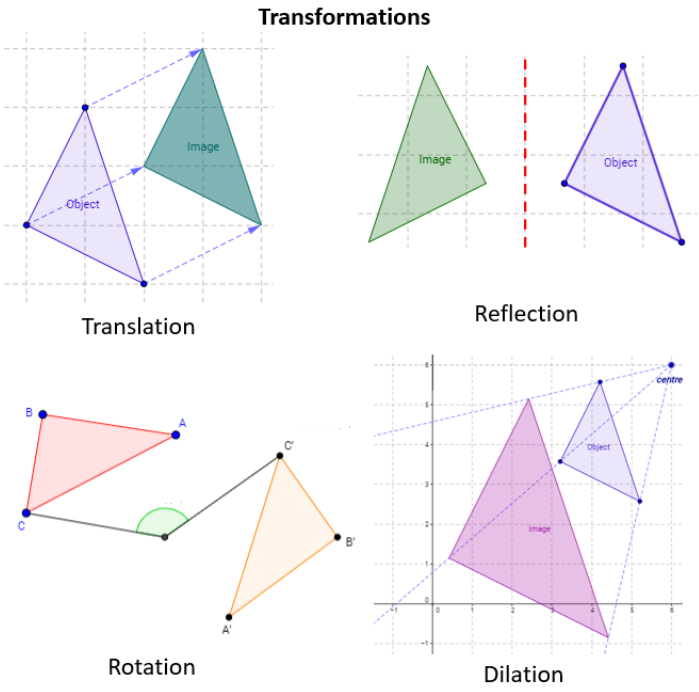



**Figure 2-5:** Isometric Transformations. A transformation is considered isometric if it does not change the length, area, or relative angles between any pair of points. The image shows how translations, rotations and reflections satisfy the isometry definition, and how dilations are examples of non-isometric transformations. Source: `https://www.onlinemathlearning.com/math-transformation.html`

.

**Point clouds** a point cloud is a set of data points in a euclidean space. Point clouds are generally produced by 3D scanners, which measure a large number of points on the external surfaces of objects around them. As the output of 3D scanning processes, point clouds are used for many purposes, such as create 3D CAD models for manufactured parts, for metrology and quality inspection, and a multitude of visualization, animation, rendering, and mass customization applications.
Point clouds can also be employed to represent volumetric data, as is sometimes done in medical imaging. By using point clouds, we can achieve multi-sampling and data compression.

**Triangular Meshes** a triangular mesh is a discrete approximation of the geometric manifold of an object using a set of triangles whose that share some of their edges. They can approximate arbitrary topologies as a piecewise smooth surface.
A mesh is usually digitally stored as an array of vertices and a list of face indices. Each set of face indices consists of a tuple of three integer vertex indices. Although triangular meshes are the most common, it is possible to build a polygonal mesh out of any regular polygon. Their graph structure makes them suitable for a large variety of applications. They are also

orientable, which means that the direction of the normal to the surface in every polygon is stored as well.

**Singed Distance Functions** a signed distance function of a set $\Omega$ is a function $f$ that determines the distance of a given point $x$ from the boundary of $\Omega$. The sign of $f$ accounts for the point $x$ being inside or outside $\Omega$. Positive values mean that $x$ is inside $\Omega$. It decreases in value as $x$ approaches $\delta\Omega$, where the signed distance function is zero, and it takes negative values outside of $\Omega$. Formally,

$$f(x) = \begin{cases} d(x, \partial\Omega) & \text{if } x \in \Omega \\ -d(x, \partial\Omega) & \text{if } x \in \Omega^c \end{cases}. \tag{2-4}$$

It is relevant to mention that the opposite convention is also adopted sometimes.

## 2.7 Medial Axis Transform

The medial axis transform (MAT) of an object $\Omega$ is the set of all pairs $(x, r)$ such that a ball $B_r(x)$ centered on $x$, with radius $r$, and totally inscribed in $\Omega$ is not contained in any other ball centered on $x$ as well. This means $B_r(x) \nsubseteq B_{r'}(x)$, for all $r' > r$.
Several alternative definitions of the MAT exit in the scientific literature. One of the most widely used defines the MAT as the set of all points having more than one closest point on the object's boundary. Both of the previous definitions are equivalent and lead to the same result.
It is important to make the distinction between the Medial Axis Transform and the Medial Axis. The medial axis only contains the locus of the inscribed balls and not the radius information. The MAT, instead, is a complete shape descriptor, meaning it can be used to reconstruct the shape of the original domain. In the 3D case, the medial axis transform is also known as the medial surface. See figures **2-6** and **2-7**.
The MAT was originally referred to as the topological skeleton. It was introduced by Blum (Blum, 1967) as a tool for biological shape recognition. It is used as a shape representation of low dimensionality, but with all the topological information of the object contained in it. Moreover, it is easy to see from its definition that the medial axis exhibits invariance to isometric transformations.
Although there is no unanimous agreement among authors, the media axis transform should satisfy five main properties: 1) have the same topology as the object, i.e., the same number of components and holes (and tunnels); 2) be thin; 3) be centered within the object (medial); 4) preserve the geometric features of the object; and 5) allow complete recovery of the original object (Punam K. Saha and de Baja , Eds.).

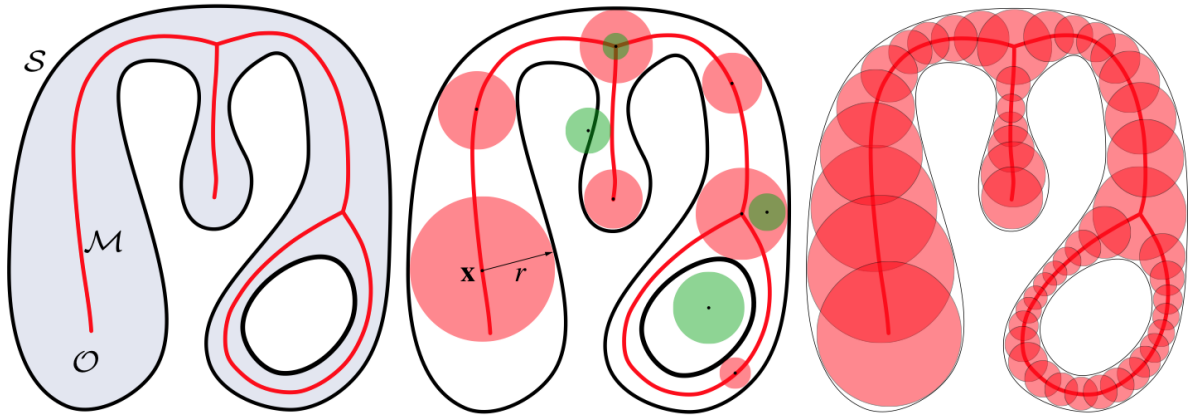



**Figure 2-6:** Illustration of the medial axis definition. (left) The MAT skeleton $\mathcal{M}$ of the shape $\mathcal{O}$ with contour $\mathcal{S}$. (middle) Examples of maximally inscribed balls (red), and balls which are neither maximal nor inscribed, thus not contributing to $\mathcal{M}$ (green). (right) Approximate reconstruction of $\mathcal{O}$ by the union of balls $B_r(x)$. Source: (Tagliasacchi et al., 2016)

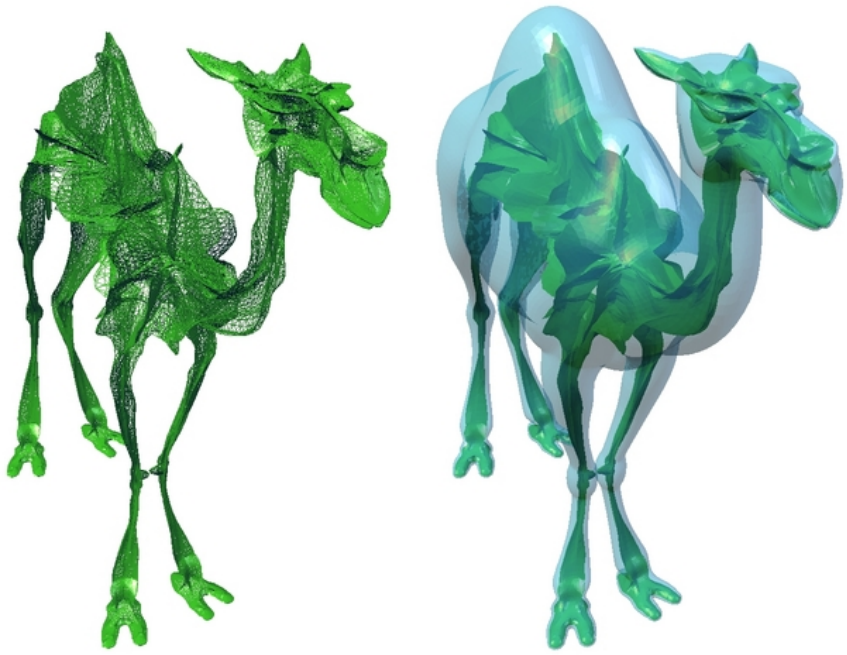


**Figure 2-7:** Medial Axis of a 3D object. Medial axis surface of the objecet (left). Medial axis surface over imposed on the original object (right). Source: `http://www2.riken.jp/brict/Yoshizawa/Research/Skeleton.html`

Because the MAT is one of the main focuses of our dissertation, we will expand its definition and explore its properties in detail in subsequent chapters.

# 3 Literature Review

In this chapter, we offer a short literature review of shape representations and shape descriptors. Despite the difficulty of categorizing the extensive corpus of studies in the field, we have chosen an intuitive way to group them into the following five categories: 1) contour-based, 2) region-based, 3) graph-based, 4) spectral shape description, and 5) learned shape description. In the following sections, we will present an overview of each one of these categories with their advantages, weaknesses, and illustrative examples.

## 3.1 Contour and Surface Based Shape Description

The contour of a 2D object is a closed curve that contains essential information to understand the object geometry. For example, humans can recognize an object solely by its contour. Instead of a closed curved, a 3D object is usually represented as an oriented surface. Such surface accounts for the boundary of the object in $\mathbb{R}^3$. For simplicity, we will refer here to both 2D or 3D approaches as contour-based shape descriptors. However, we will offer additional clarification when necessary.

Contour-based descriptors (CBD) only consider the boundary of the shape and neglect the information contained in the shape interior. These descriptors are very efficient at filtering out results based on the boundary points because of their low computation complexity. However, they are not good at handling image noise, and thus they are not accurate in real-life applications. All contour-based shape description methods depend on a parametrization of the contour. If this parametrization is too coarse or the curve is not smooth enough, the performance can drop (Stiene et al., 2006) significantly.

Perhaps the simplest shape descriptor is the chain codes (Liu and Žalik, 2005). Chain codes describe the boundary of a 2D object by encoding the angles between a point in the boundary and its immediate neighbors. Different encodings are obtained by using either a 4-element or 8-element neighborhood. Each element in the chain encodes as a relative angle difference between itself and the next element. After, some statistics are computed, such as the number of times a code repeats. These statistics are later used as features for shape comparison. Chain codes are invariant to translation, but not to scale neither rotation.

Another widely-used shape descriptor is the Shape Context (SC) (Belongie et al., 2002). The SC is a way of capturing the relationship between a single point and its neighbors in a uniform radial vicinity. Given a shape $\Omega$, SC defines a polar histogram on each point of its boundary $\delta\Omega$. The histogram is the descriptor itself.

Histograms of different points in different shapes are compared to do shape matching. There are many variations of the original SC formulation (Bhuptani and Talati, 2014; Kokkinos et al., 2012) intending to make it less computational expensive (Mori et al., 2005), and invariant to rotations (Yang and Wang, 2007) and deformations (Ling et al., 2007).

Another well-known CBD is the Fourier descriptor (FD). FD is the result of parametrizing the contour as a Jordan Curve $r(t)$, and applying the Fourier transform. The descriptor is defined as vector $v \in \mathbb{R}^m$, where the entries of $v$ are the first $m$ coefficients of the Fourier Representation of the shape signature of the object. Here, shape signature means any 1-D function used to represent the boundaries of a 2-D shape,

$$FD(\Omega) = a_0, a_1, a_i, ..., a_{m-1}, \text{ with} \qquad a_i = \frac{1}{n}\sum_{t=0}^{n-1} r(t) e^{\frac{-(j2\pi nt)}{n}} \tag{3-1}$$

Despite its simplicity, FD is a powerful descriptor that has invariance to translations, and equivariance to rotations and scale. However, FD's representation performance reduces when the contour is undersampled. Variations to FD include (Wafi et al., 2016), (Zhao and Belkasim, 2012), and (Ye Mei and Androutsos, 2008).

The parametrized curve $r(t)$ is also employed to define another shape descriptor: The Curvature Scale (Abbasi et al., 1999) (CSS). The CSS is computed using a 2D plot called the CSS image with dimensions $(t, \sigma)$. Every point in this plot corresponds to a location of curvature zero-crossing along with the parameter $t$, on a Gaussian-smoothed version of $r(t)$. The zero-crossing is estimated as

$$k(t,\sigma) = \frac{X_t(t,\sigma)Y_{tt}(t,\sigma) - X_{tt}(t,\sigma)Y_t(t,\sigma)}{(X_t(t,\sigma)^2 + Y_t(t,\sigma)^2)^{\frac{3}{2}}}. \tag{3-2}$$

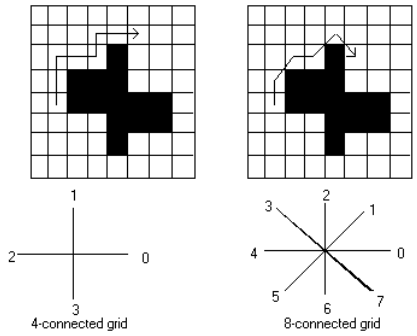
The curve along the parameter $t$, for a range of values of $\sigma$ (starting at 1) is defined as:

$$r_\sigma(t,\sigma) = [X(t,\sigma), Y(t,\sigma)]^T \tag{3-3}$$

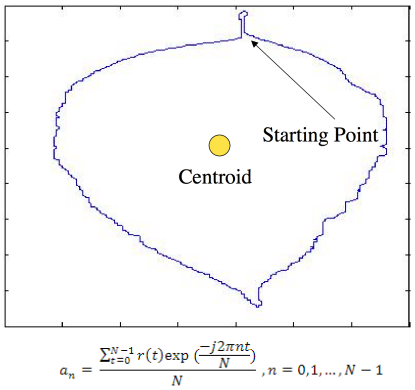
The CSS descriptor consists of the locations of the maxima of the CSS image on the plane $(t, \sigma)$. The CSS representation is robust with respect to scale, noise, and change in orientation. Any rotation of the object causes a circular shift on its representation, which is easily determined during the matching process.

A number of other contour-based approaches describing features of shapes do exist in the scientific literature applied to different fields. Most of the work, however, has been done in the area of free form object recognition and classification in 3D range data. Examples of this works include eigen-CSS (Drew et al., 2009), B-Spline contour descriptors (Figueiredo et al., 2000), and geometric metrics such as (roundness, relation Area/Perimeter, mean curvature,

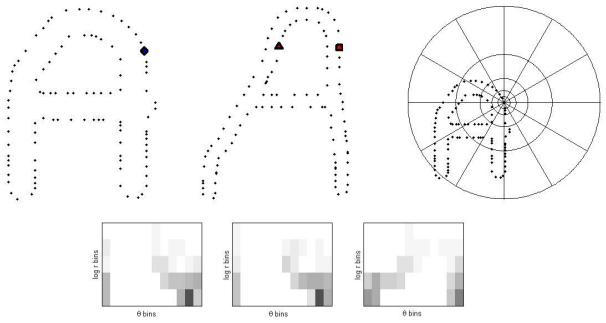
etc.), or the Chordiogram (Toshev, 2011; Toshev et al., 2012) that we will discuss in detail later in this study.



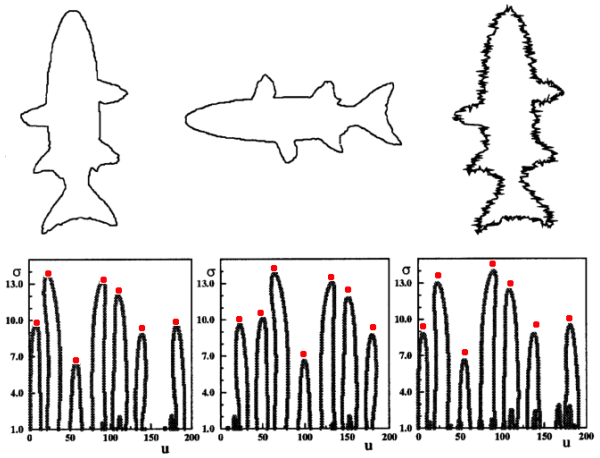

**(a)** Chain Codes. The k-neighborhood define the encoding of the shape. Statistics of the codes are use as feature vector. For the examples shown, the codes are 1100100..., and 21017...




**(b)** Fourier Descriptors. The contour of the object is parametrized as Jordan Curve, $s(t)$, and the coefficients of the Fourier Transform define the descriptor.



**(c)** Shape Context. Each point in the contour is described through a 2D polar histogram. Similar points have similar histograms.




**(d)** The maxima of curvature scale space. Consist of the locations of the maxima of the CSS image, constructed by evolving the contour of an object with successive Gaussian smoothing.

**Figure 3-1:** Examples of the four most relevant contour-based shape descriptors.

## 3.2 Region Based Shape Description

Region-based methods (RBD) take all of the pixels within a shape region into consideration to obtain the shape descriptor. They further divide into global and local region-based methods. Global methods take the region as a whole during the calculation. Local methods divide the region into smaller parts called primitives and then accumulate the results together at the end. The most popular RBDs are different “moments” (regular moments, Zernike moments (ZMD), Hu-moments), Angular Radial Transform (ART), and the popular Scale Invariance Feature Transform (SIFT) descriptor (Lowe, 2004).

In general, moments describe numeric quantities at some distance from a reference point or axis. Regular moments, for instance, have simple transnational and scale-invariant properties. However, the basis ($x^p$ and $y^q$) are not orthogonal, and therefore, regular moments contain redundant information. All regular moments have the form:

$$M_{pq} = \int_{\infty}^{\infty} \int_{\infty}^{\infty} x^p y^q f(x,y) dx dy, \tag{3-4}$$

Where $p$ and $q$ are non-negative integers, and $f(x,y)$ is the image function whose moments we want to compute. In computer vision, $f(x,y)$ is usually a binary image representing an object segmentation.

Moments produced using orthogonal basis sets, like Zernike Moments (Khotanzad and Hong, 1990), better encode the information of the image and require lower computational precision to the same accuracy. Zernike Moments are a set of orthogonal polynomials defined on the unit disk. They have simple rotation and scale invariance, and higher accuracy for detailed shapes. The basis for their computations are the polynomials:

$$\begin{aligned} &V_{nm} = R_{nm}(\rho) e^{jm\theta}, \text{with} \\ &R_n^m = \begin{cases} \sum_{l=0}^{n-m/2} \frac{(-1)^l (n-l)!}{l![\frac{1}{2}(n+m)-l]![\frac{1}{2}(n-m)-l]!} \rho^{n-2l} & \text{for n - m even} \\ 0 & \text{for n - m odd} \end{cases} \end{aligned} \tag{3-5}$$

Additionally, Hu-moments (Zhihu Huang and Jinsong Leng, 2010) are defined a set of 7 formulas, $h_i$, that are basically a non-linear combination of regular moments. They exhibit scale, rotation and translation invariant properties. e.g., $h_2 = (M_{20} - M_{02})^2 + 4M_{11}^2$.

The Angular Radial Transform (Kim and Kim, 2000) (ART) is another moment-based descriptor for both connected and disconnected shapes. The ART is a complex orthogonal, unitary transform defined on a unit disk that consists of the complete orthogonal sinusoidal basis functions in polar coordinates. The ART coefficients, $F_{nm}$ of order $n$ and $m$, are defined as

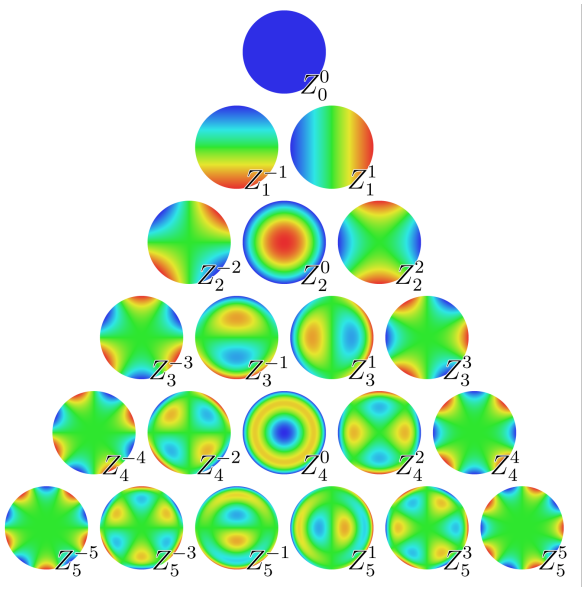
$$F_{nm} = \int_0^{2\pi} \int_0^1 V_{nm}(\rho,\theta) f(\rho,\theta) \rho d\rho d\theta, \tag{3-6}$$

Where $f(\rho,\theta)$ is the image in polar coordinates, and $V_{nm}(\rho,\theta) = A_m(\theta) R_n(\theta)$ is the ART basis. In order to achieve rotation invariance, an exponential function is used for the angular

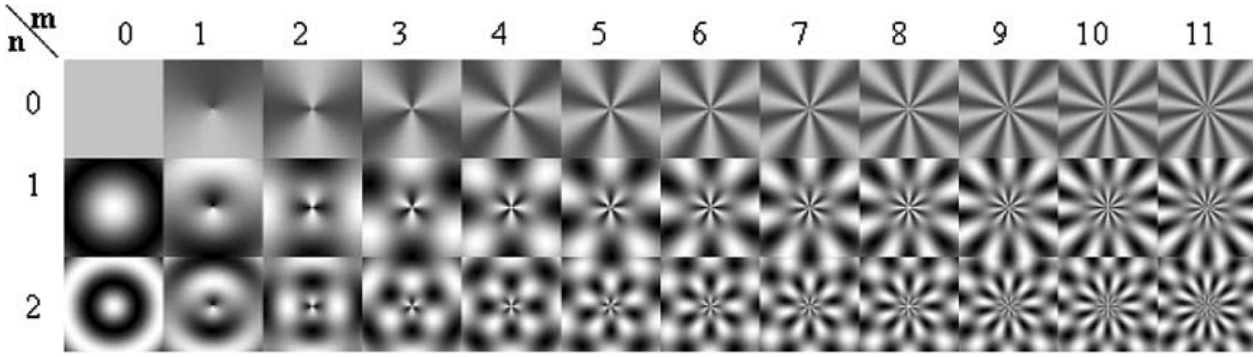
basis function. See equation 3-7. Real parts of basis functions are shown in Figure **3-2**b.

$$A_m(\theta) = \frac{1}{2\pi} e^{jm\theta}$$
$$R_n = \begin{cases} 1 & n = 0 \\ 2cos(\pi n \rho) & n \neq 0 \end{cases} \tag{3-7}$$

A significant number of other contour-based approaches describing features of shapes exist in the scientific literature. Examples of this work include Lowe's (Lowe, 2004) SIFT for RGB images, geometric metrics (area-perimeter ratio, roundness, Danielsson factor, etc.), or 3DMatch (Zeng et al., 2016) for 3D data.




**(a)** The first 21 Zernike polynomials, ordered vertically by radial degree and horizontally by azimuthal degree.




**(b)** Real parts of the The Angular Radial Transform basis functions.

**Figure 3-2:** Examples of two of the most relevant region-based shape descriptors.

## 3.3 Graph Based Shape Description

In this section, we will explore description methods intended to represent the geometric properties of the object as a graph. The main reasons for employing this methodology are: 1) the fact that a graph representation is usually of lower dimensionality, and 2) that most of the graph-based methods preserve the topology (genus) of the objects. Despite the ambiguous definition of graph-based descriptors, graph-based usually means that the features are computed from the Medial Axis Transform (MAT) of the object. The MAT, also known as the topological skeleton, was defined by (Blum, 1967) as the set of all points having more than one closest point on the object's boundary. The MAT is usually presented as tuples of the form $(X, r)$. Where $X \in \mathbb{R}^n$ is the location of the skeleton point inside the object, and $r \in \mathbb{R}^+$ is the radius of a ball centered on $X$. This representation allows the reconstruction of the original object from its MAT.

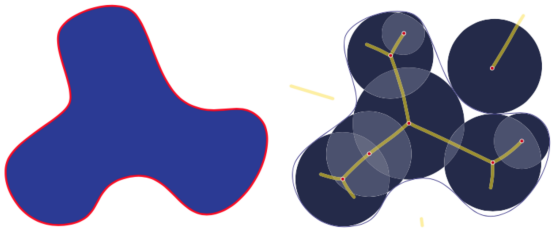
A large number of shape descriptors have been proposed by exploiting the properties of the MAT or re-defining its computation. There are three primary mechanisms to compute the MAT, also called skeletonization techniques: 1) the layer by layer erosion (also called thinning) method, 2) calculating the Voronoi diagram generated by the boundary points, and 3) detecting ridges in distance map of the boundary points. In digital spaces, only an approximation to the"true skeleton" can be extracted.

In the process of thinning to obtain the medial axis of an object, the points belonging to its volume (or silhouette in 2D), are deleted from the outer boundary first and then proceeded inside until a single-pixel wide skeleton result. These methods are easy to implement; however, they are not robust to isometries. Skeletonization by thinning can be expressed in terms of morphological erosions and openings:

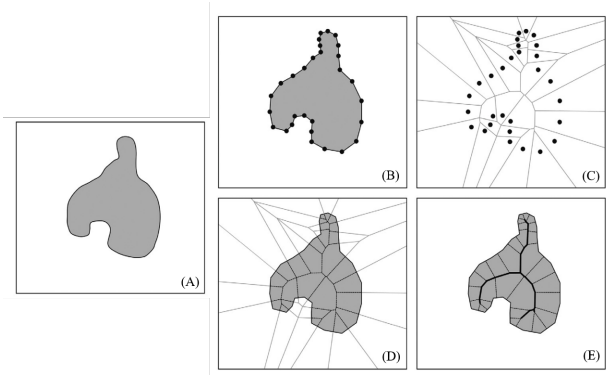
$$S(A) = \bigcup_{k=0}^{K} S_k(A), \text{ with } S_k = \bigcup_{k=0}^{K} \{(A \ominus kB) - [(A \ominus kB) \circ B]\} \tag{3-8}$$

Where $B$ is a structuring element, $(A \ominus kB)$ indicates k successive erosions of $A$, and $K$ is the number of iterations required before $A$ becomes an empty set. The most well-known algorithm for thinning skeletonization is perhaps the Zhang Suen (Zhang and Suen, 1984) algorithm; however, other approaches have been developed using similar principles (Viswanathan et al., 2013). The skeletonization can fulfill both property requirements (i.e., the topological and the geometrical) based on Voronoi diagrams. Nevertheless, this is an expensive process, especially for large and complex objects.

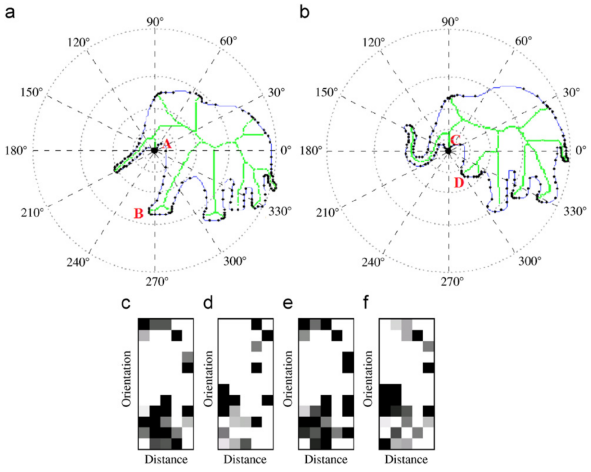
Skeletonization can also be estimated by taking sample points on the contour of the object and then compute the Voronoi Diagram (VD) of such points (Ogniewicz and Ilg, 1992). The skeleton is then the intersection of the VD with the object itself. One disadvantage of this method is that each additional vertex on the polygon adds a new skeletal branch. Thus, a



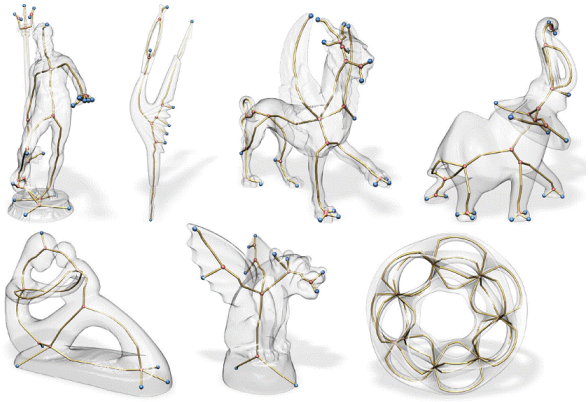
**(a)** Medial Axis Transform Computation. (Left) a shape and its boundary. (right) Medial Axis elements consisting of the centers and radius of balls inscribed in the shape (Peters and Ledoux, 2016)




**(b)** Computation of a Voronoi skeleton. The last step (bottom right) shows some examples of spurious branches. Saha et al. (2016)




**(c)** Skeletal context. This descriptor uses a similar formulation as the shape context; using points from the contour that depend on the branches of the medial axis (Xie et al., 2008).



**(d)** 3D skeletonization example (Au et al., 2008)

**Figure 3-3:** Examples of computations of the Medial Axis Transform in 2D and 3D

suitable polygonal approximation of an object is crucial to generate the desired complexity of the skeleton (Punam K. Saha and de Baja , Eds.).

The most common methods to extract the MAT are, however, those based on the distance transform. In these methods, the skeleton is calculated as the ridges of the distance transform inside the silhouette/volume of the object. Some authors refer to this approach as curve evolution or grass fire transform since this process can be described as "setting a fire" on the borders of an image region to yield descriptors such as the region's skeleton or medial axis.

A factor that can limit the use of the skeleton in applications is its sensitivity to noise along the object boundary (Beristain and Grana, 2010). Even negligible boundary noise can cause spurious skeleton branches, so that skeleton pruning techniques are of interest. Effective pruning techniques focus on different criteria for the evaluation of the significance of an individual skeleton branch. Thus, the decision is whether to remove or keep the branch. Clearly, branch removal by pruning modifies the skeleton in such a way that a smoother

boundary characterizes the object represented by the pruned skeleton. A pruning process is adequate if the resulting skeleton structure is noticeably simplified, but the above differences are negligible for the specific application. Some authors address this issue by including constraints while computing all the points that belong to the medial axis. Others do so by removing branches that are considered useless according to criteria like the reconstruction accuracy(Punam K. Saha and de Baja , Eds.; Gao et al., 2018; Hesselink and Roerdink, 2008).

Alone, it is difficult to use the Medial Axis Transform as a shape descriptor, because the number of tuples for each object might be different. It is, instead, considered a simplified shape representation. Hence, many works have explored extracting a certain number of features based on the MAT; and then using these as a descriptor for shape classification and shape retrieval. Descriptors based on the MAT usually incorporate features from other methodologies not strictly associated with shape. One example is the Bag of Skeleton Paths (Shen et al., 2014a). This descriptor is computed by pooling the skeleton paths connecting pairs of endpoints in the skeleton, in a bag-of-words fashion. Another example is the Skeletal Context (Xie et al., 2008), which is based on the well-known Shape Context discussed in section 3.1. Skeletal Context aims to build a histogram of points on the contour of the object, as SC does, but not using a regular sampling of the contour of the object. Instead, it uses the endpoints where the medial axis's branches touch the contour. See figure **3-3**c.

Ongoing research related to skeletonization is moving on two fronts in recent years. On the one hand, the traditional approach of using a segmented silhouette of an object to extract the skeleton has lost popularity compared to approaches that compute the skeleton on natural-colored images with complex backgrounds (Tsogkas and Dickinson, 2017). On the other hand, deep learning methods have been developed using geometric constraints and reconstructions constraints into the loss function (Atienza, 2019; Wang et al., 2018). These geometric constraints are direct applications of the medial axis properties discussed above.

## 3.4 Spectral Shape Analysis

Spectral shape analysis is a new but exciting field. It describes and compares geometric shapes based on the spectrum (eigenvalues or eigenfunctions) of the Laplace–Beltrami operator. The Laplace-Beltrami operator is the generalized form of the Laplacian operator $\Delta$ sometimes denoted as $(\nabla^2)$. Like the Laplacian, the Laplace–Beltrami operator is defined as the divergence of the gradient and is a linear operator taking functions into functions.

$$\Delta f = (\nabla.\nabla)f = \nabla^2 f = div(\nabla f). \tag{3-9}$$

Since the spectrum of the Laplace–Beltrami operator is invariant under isometries, it is well suited for the analysis of the shape of objects under arbitrary rotations, as well as

for classification and retrieval of non-rigid shapes, i.e., bendable objects such as humans, animals, plants, etc.
The spectral components of the Laplace-Beltrami operator can be computed by solving the Helmholtz partial differential equation (or Laplacian eigenvalue problem):

$$(\nabla^2 + \lambda)f = 0. \tag{3-10}$$

The solutions are the eigenfunctions $\phi_i$ and eigenvalues $\lambda_i$ of operator $\nabla$. The structure of the eigenfunctions depends on the geometry of the manifold, where the partial differential equation is solved. For instance, for a sphere in 3D ($S^2$), the eigenfunctions turn out to be the spherical harmonics.
Geometric shapes are often represented as 2D curved surfaces, 2D surface meshes (usually triangle meshes), or 3D solid objects (e.g., using voxels or tetrahedra meshes). The Helmholtz equation can be discretely solved for all these cases. If a boundary exists, e.g., a square, or the volume of any 3D geometric shape, boundary conditions need to be specified.
Several discretizations of the Laplace operator exist for the different types of geometry representations. However, some of these operators do not approximate the underlying continuous operator well enough and should be used carefully.
The recent interest of the scientific community in spectral analysis has resulted in a considerable number of spectral shape signatures that have been successfully applied to a broad range of areas, including manifold learning (Belkin et al., 2006), object recognition and deformable shape analysis (Pickup et al., 2016a; Li and Ben Hamza, 2013), medical imaging (Chaudhari et al., 2014), and shape classification and retrieval (Gao et al., 2014). The diversified nature of these applications demonstrates the practicality of spectral analysis.
The simplest, albeit still useful descriptor extracted from the Laplace-Beltrami operator is called Shape-DNA (Reuter et al., 2006). Shape-DNA consist of using the cropped spectrum containing only the first $n$ eigenvalues. Assuming $\mathcal{M}$ to be a *Riemannian Manifold* with metric $h$, Shape-DNA is defined as:

$$ShapeDNA(\mathcal{M}, h) = [\lambda_0, \lambda_0, ..., \lambda_{n-1}]^T, \text{ with } \lambda_i \leq \lambda_{i+1} \tag{3-11}$$

Its main advantages are its simple representation (a vector of numbers) and comparison, its scale invariance, and its good performance for shape retrieval of non-rigid shapes, despite its simplicity. There are several formulations of shape descriptors theoretically close to ShapeDNA: singular values of Geodesic Distance Matrix (SD-GDM) (Smeets et al., 2009) and Reduced Bi-Harmonic Distance Matrix (R-BiHDM) (Ye and Yu, 2015). However, the eigenvalues are global descriptors, and therefore the shapeDNA and other global spectral descriptors cannot be used for local or partial shape analysis.
The eigenfunctions of $\nabla^2$ can be used to perform shape analysis as well. The most common

types of these shape descriptors are called *Signatures*. Notable examples in this family are the heat kernel signature (HKS) (Sun et al., 2009) and the wave kernel signature (WKS) (Aubry et al., 2011); however, other variations like Global point signature (GPS) (Rustamov, 2007), Improved wave kernel signature (IWKS) (Limberger and Wilson, 2015), and Spectral graph wavelet signature (SGWS) (Masoumi et al., 2016) also exist. HKS is based on the concept of heat diffusion over a surface. Given an initial heat distribution $u_0(x)$ over the surface, the heat kernel $h_t(x, y)$ relates the amount of heat transferred from $x$ to $y$ after $t$. The solution to the diffusion equation:

$$\Delta u(x,t) = k\frac{\partial u(x,t)}{\partial t} \tag{3-12}$$

Can be express in terms of the eigenvalues and eigenfunctions of $\Delta$:

$$u(x,t) = \int h_t(x,y)u_0(y)dy \tag{3-13}$$

with,

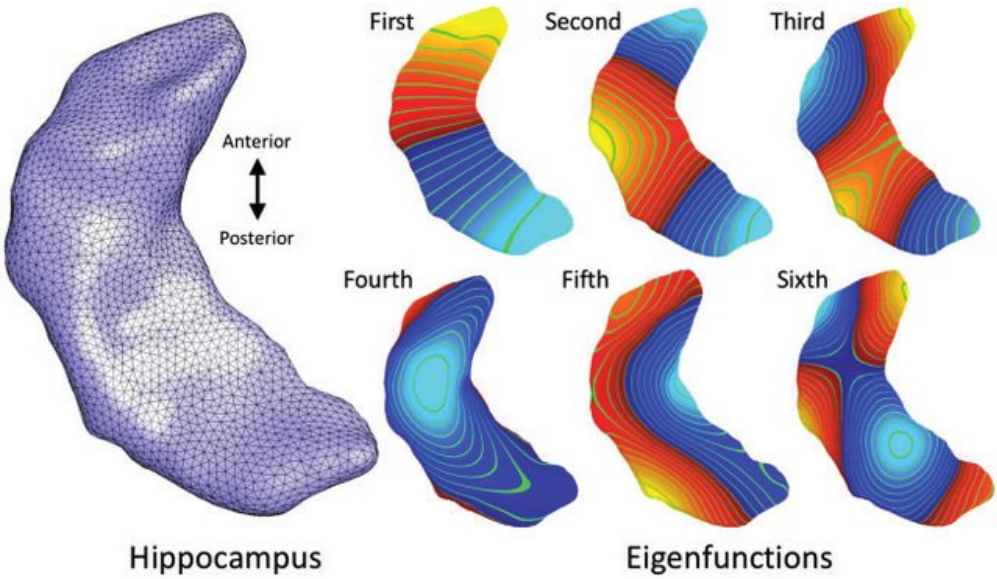
$$h_t(x,y) = \sum_{i=0}^{\infty} e^{-\lambda_i t}\phi_i(x)\phi_i(y) \tag{3-14}$$

The heat kernel fully characterizes shapes up to an isometry and represents increasingly global properties of the shape with increasing time. The constant $k$ in equation 3-12 stands for the material diffusivity.

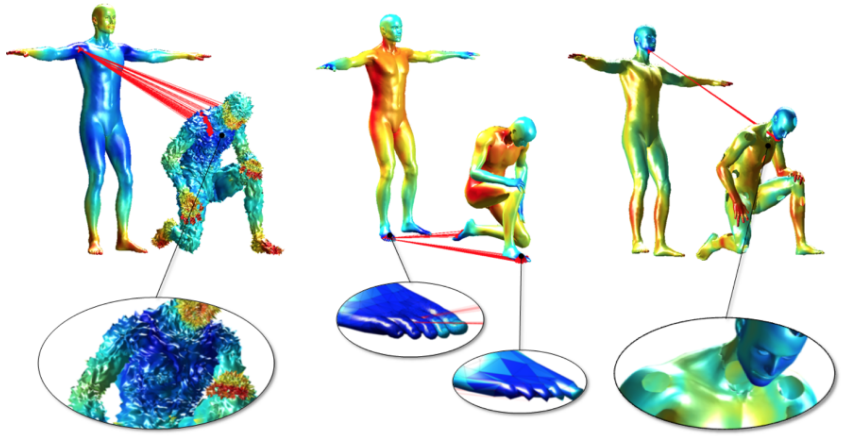
Spectral shape analysis is still an emerging topic. Part of the ongoing research in this area focuses on incorporate concepts from Deep Learning. Hence, some work has been done on trying to shift the properties of the spectrum of the manifold of a shape into a Machine Learning framework. The spectrum of different manifolds is, in general, different (eigenvalues and eigenfunctions differ between shapes). Therefore, models like the one in (Litman and Bronstein, 2014) propose to model the intrinsic information provided for the spectrum as a sort of dictionary of basis $b(\lambda_i)$. Generalizing equation 3-14, and re-writing it we can express a spectral descriptor as:

$$p(x) = \sum_{k\geq 1} f(\lambda_k)\phi_k^2(x) = \sum_{k\geq 1} Ab(\lambda_k)\phi_k^2(x) = Ag(x) \tag{3-15}$$

Where $g_j(x) = \sum_{k\geq 1} b_j(\lambda_k)\phi_k^2(x)$ stores all the geometric information generalized across a dictionary of functions of an $m >> n$ number of frequencies $b = \{b_j(\lambda), ..., b_m(\lambda)\}$. Equation




**(a)** A 3D shape (left) and its six first eigenfunctions plotted over it. Taken from (Wachinger et al., 2016).



**(b)** Robustness of the WKS: The red lines connect a reference point in the background (standing David) with its 50 best matches on the perturbed shape in the foreground. The color encodes the feature distance to the reference point, blue = short distance, red = large distance in the feature space.

**Figure 3-4:** Examples of how the eigenfunctions look when plotted on the shape manifold (left). The WKS descriptor used for point correspondence (right).

3-15 leads to the formulation of the cost function

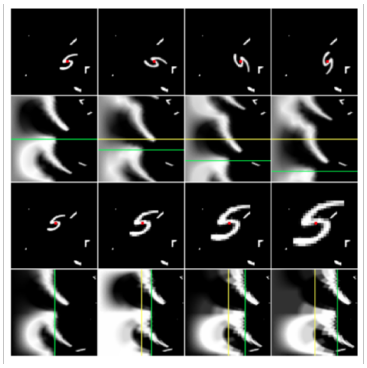
$$d^2 = ||p - p'||^2 = ||A(g - g')||^2 = (g - g')^T A^T A(g - g'), \tag{3-16}$$

that is suitable for a machine learning approach. The matrix $A^T A$ contains the weights to be learned while minimizing $d^2$. $A$ represents the coefficients of the response to any of the

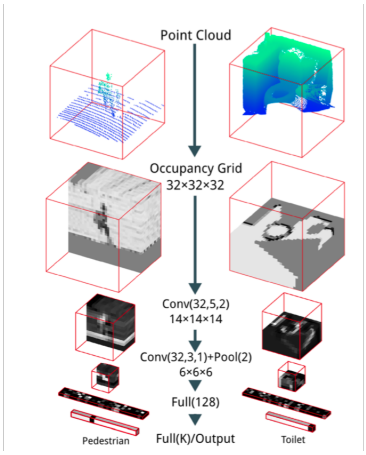
frequencies in $b$.
In the next section, we will discuss more recent shape descriptors based on machine learning and deep learning. Some of these descriptors also incorporate concepts from spectral shape analysis.

## 3.5 Learned Shape Features

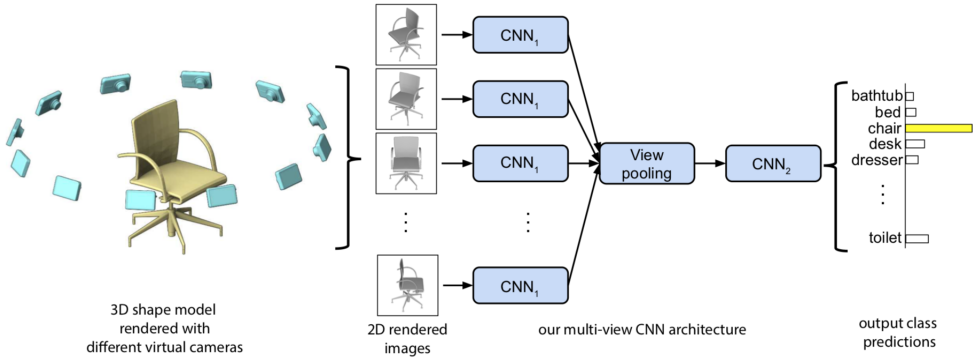
Deep Learning has an astonishing impact in science and modern life. Deep Learning approaches have a wide variety of applications across numerous fields. Thus, it is unsurprising that Shape Analysis is one of the fields in which Deep Learning has an impact.
There are many learned models for 3D shapes expressed as point clouds (Qi et al., 2017a), or triangular meshes (Kulon et al., 2019), but very few for 2D shape analysis. In Atabay (2017), the authors present an interesting comparison of several CCN models used for binary Shape Classification. This paper shows that CNN's models have competitive, and many times higher, performance compared with hand-crafted features over the most common binary shape dataset (e.g., Animal Dataset, MPEG7, ModelNet (Zhirong Wu et al., 2015), and ShapeNet (Chang et al., 2015)).
The goal of Deep Learning for Shape Description is to let a machine learning system to learn the filters/models that offer a useful response to the geometry of an object. Also, learned descriptors should have the capacity to generalize through different shapes, so that only a single model needs to be trained.
Despite the high accuracy achieved by these studies, a few challenges remain open problems in computer vision. Perhaps the most notorious one is the inability of CNN's to properly response to natural rotations of the object. CNN's are not invariant to rotations (Chui et al., 2019). This is an especially difficult problem in 3D, when dealing with arbitrary rotations on the rotation mathematical group $\mathbf{SO}(3)$ (Esteves et al., 2018a). Some authors have developed exciting approaches where good accuracy is achieved in the presence of arbitrary rotations. Most of them, however, use very large deep learning models, which lead to the discussion of how much the network is learning or memorizing a subset of all the possible rotations.
Grid-based approaches to Learned Shaped descriptions are the most straightforward way to use deep learning for shape analysis. In this sense, a CNN is applied to the image or volume, hoping that the network will learn characteristic enough features of the geometry of the objects in the dataset (Maturana and Scherer, 2015; Wang et al., 2019a). These approaches are simple; however, they also suffer from lack of rotation invariance and are computationally expensive because they depend exponentially on the tessellation of the object. The finer the grid is, the more complex and time consuming the computations are.
Multiview CNN (MVCNN) (Su et al., 2015; Feng et al., 2018) are perhaps the most representative deep learning model for 3D shape classification that suffers from the stated rotation non-invariant problem. MVCNN, in essence, train a CNN on a large set of rendered views of the object (usually between 12 and 80 views). These views are passed through a regular



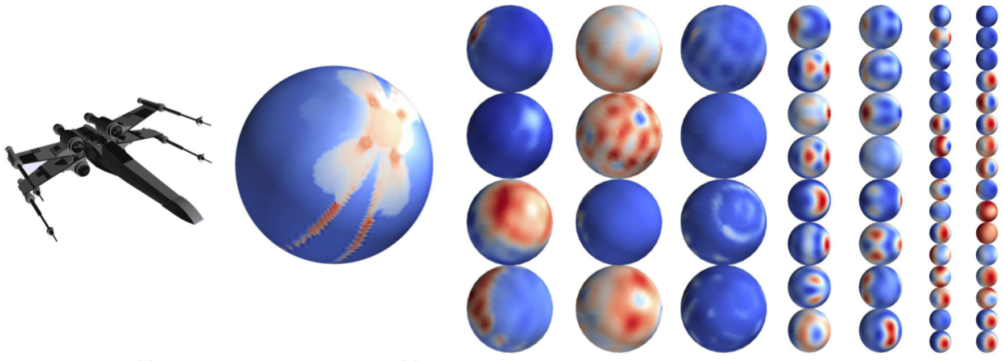
**(a)** In the log-polar representation, rotations around the origin become vertical shifts, and dilations around the origin become horizontal shifts (Esteves et al., 2018b).




**(b)** VoxNet architecture. The network voxelize the object's point cloud in order to apply 3D convolutions on a regular grid.




**(c)** Multiview CNN depicting the process of view rendering, and how each view is processed independently by a CNN before pooling into the final features vector (Su et al., 2015).



**(d)** Spherical CNNs. (Left) A 3D object. (Center) The projection of the 3D object into a spherical function. (Right) Spherical features learned by the neural network.

**Figure 3-5:** Examples of learned features describing different methods to achieve rotation equivariance using machine learning approaches.

CNN, then pooled together, and finally passed through a fully connected network to produce the final features for classification. MVCNN has over 99 million parameters. This makes it hard to train and also poses the "memorizing problem" discussed above.

A solution to this problem is to create equivariant CNN models where the learned features of a rotated object are the rotated features of the original one. Recall equation 2-1). Spherical CNNs (Cohen et al., 2018; Esteves et al., 2018a) are one example of equivariance networks. The main idea behind them is to represent 3D objects as functions on the sphere, and then run a series of convolution layers on such representation. Due to the non-euclidean geometry of the sphere, the convolutions are computed not with a fast sliding window approach, but with spherical convolutions in the Fourier domain via spherical harmonics.
Polar Transformer Network (Esteves et al., 2018b) offers a solution to the rotation invariance problem in 2D. This type of CNN learns the transformation between a 2D image into a polar image. In log-polar coordinates, a rotation of the original object becomes a translation; and then, a regular CNN learns the features of the object. Regular CNN's are equivariant to translations by design due to the well-known properties of convolutions.
Dealing with non-rigid transformation in shapes is also a central problem in computer vision. Most shape descriptors, however, are only invariant up to isometries. Thus, changes in the object geometry like articulations or general changes due to movement or 'èvolution" of the object are not covered. Nevertheless, there are some exceptions. Graph-based descriptors are invariant to articulations due to the graph structure that models the joints (Pickup et al., 2016b). Spectral shape descriptors offer some degree of invariance to small deformation as long as they are locally enough. This means that if the deformation occurs in a small neighborhood of a point where the manifold of the object does not change abruptly, the descriptor remains relatively stable.
As a consequence, a subset of learned shape analysis is conducted by integrating ideas from graph-based and spectral techniques into a machine learning pipeline. This is due by engineering loss functions that take into account the graph representation of the object Pumarola et al. (2018), or by incorporating the spectral analysis into the process of learning the filters on the shape manifold (Laga, 2018; Litany et al., 2018). A summary of learned approaches to shape analysis is shown in figure **3-5**.

# 4 Shape Description Based on the Isometric Invariances of Topological Skeletonization

## 4.1 Overview of the Stated Problem

In this dissertation, we study the problem of describing the shape of an object in 2D and 3D with a set of features invariant to isometric transformations, particularly rotations. We focus our approach on the well-known Medial Axis Transform (MAT) and its topological properties. A set of experiments conducted on popular datasets was performed to highlight how our descriptor behaves compared with state-of-the-art approaches.

We aim to study two problems. The first problem: how to find a shape representation of a segmented object that exhibits rotation, translation, and reflection invariance. The second problem: how to build a machine learning pipeline that uses the isometric invariance of the shape representation to do both classification and retrieval. Our proposed solution demonstrates competitive results compared to state-of-the-art approaches.

We base our descriptor on the MAT, sometimes called a topological skeleton. Accepted and well-studied properties of the medial axis include (Bernard and Manzanera, 1999):

**Homotopy** skeletons must preserve the topology of the original shapes/images.

**One-pixel thickness** skeletons should be made of one-pixel thick lines.

**Mediality** they should be positioned in the middle of shapes (with all skeleton points having the same distance from two closest points on object boundary).

**Rotation invariance** in discrete spaces, this can only be satisfied for rotation angles, which are multiples of $\frac{\pi}{2}$ but should be approximately satisfied for other angles.

**Noise immunity** skeletons should be insensitive to shape-boundary noise.

**Reconstruction** one should be capable of reconstructing the original object from the skeleton.

These properties make the MAT a suitable input to create a shape descriptor; however, several problems arise because not all skeletonization methods satisfy all of these properties at the same time. In general, skeletons based on thinning approaches preserve topology but are noise sensitive and do not allow for proper reconstruction of the original shape. These skeletons are also not invariant to rotations.

Voronoi skeletons also preserve topology and are rotation invariant, but do not include information about the thickness of the object, which makes reconstruction impossible. The Voronoi skeleton is an approximation of the real skeleton. The denser the sampling of the boundary, the better the approximation; however, a denser sampling makes the Voronoi diagram more computationally expensive.

In contrast, distance transform methods allow the reconstruction of the original object by providing the distance from every pixel in the skeleton to the boundary. Moreover, they exhibit an acceptable degree of the MAT properties listed above, but noise sensitivity remains an issue. Given this information, we selected distance transform medial axis methods as our skeletonization strategy and focused on creating a new noise-free approach to solve the contour noise problem.

Most skeletonization methods produce a skeleton that contains many more branches than desired, which can occur for several reasons. The most common reason is border noise, where every little protrusion gives rise to a skeletal branch. Unwanted parts of the skeleton are called spurious branches. Many skeletonization methods deal with spurious branches through the process of pruning, where the spurious branches are removed. In those cases, what defines a spurious part must be determined either while computing the skeleton or as a post-processing step.

To effectively classify an object, or perform any other task with features based on the object shape, the descriptor needs to be a normalized, compact form: a map $\Phi$ that take every shape $\Omega$ to the same vector space $\mathrm{R}^n$. This is not possible with skeletonization methods because the skeletons of different objects have different numbers of branches and different numbers of points, even when they belong to the same category. Consequently, we developed a strategy to extract features from the skeleton through the map $\Phi$, which we used as an input to a machine learning approach.

In this study, we sought to design an isometric invariant shape descriptor through robust skeletonization enforced by a feature extraction pipeline that exploits such invariance through a machine learning methodology. We conducted a set of classification and retrieval experiments over well-known benchmarks to validate our proposed method.

## 4.2 Methodology Design

### 4.2.1 Robust Skeletonization

Spurious branches in the MAT are generally associated with noise in the contour. Even small perturbation can cause a new branch to appear. Spurious branches create an erroneous underlying structure of the object (Figure **4-1**). We refer to any method that outputs a consistent skeleton in the presence of several degrees of noise: "robust skeletonization".
One strategy for removal of spurious branches consists of directly smoothing the shape (Rumpf and Preusser, 2002; Mokhtarian and Mackworth, 1992), which results in a **MAT** with less noise. The filtered skeleton of the unfiltered shape is later defined as the unfiltered skeleton of the filtered shape. The main drawback of this approach is that, in most cases, the resulting medial axis is not a good approximation of the real medial axis (RMA). Additionally, the smoothing procedure can potentially change the topology of the shape resulting in a different skeleton. Miklos Miklos et al. (2010) introduced a slightly different approach with the Scaled Axis Transform (SAT). The SAT involves applying a scale transformation to the distance map and computing the unfiltered medial axis of the resulting reconstructed shape. In (Postolski et al., 2014), the authors highlighted the fact that the Scale Axis Transform is not necessarily a subset of the original shape, and went beyond Miklos' work to propose a solution that guarantees a better approximation of the RMA.

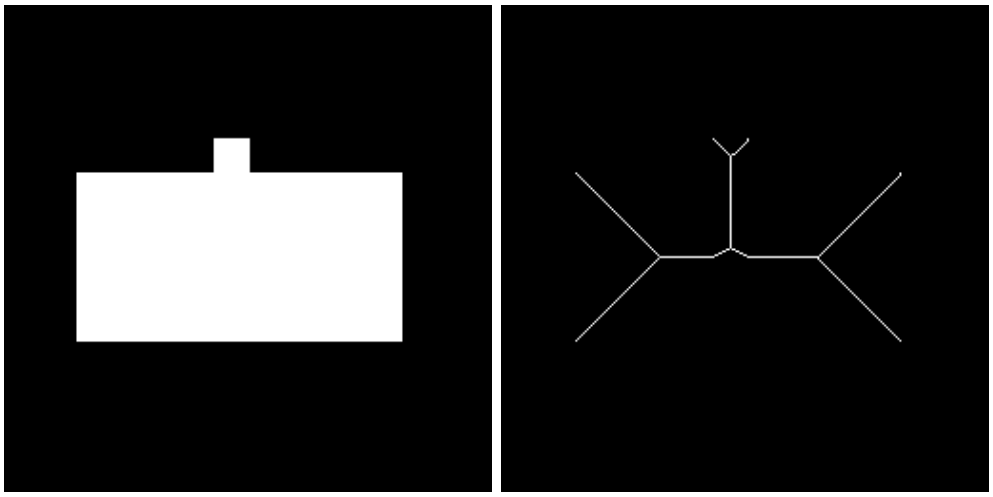


**Figure 4-1:** Spurious branch in medial axis. A new whole branch appear (right), even when a small perturbation of the contour (left).

One of the most popular solutions to spurious branches (Couprie et al., 2007) considered the angle formed by a point $p \in \Omega$, and its two closest boundary points $\Pi_\Omega(x)$, called the distance transform of $p$. This solution removes skeleton points for which this angle is lower than a fixed threshold. This criterion allows different scales within a shape but generally leads to an unconnected medial axis. Some modifications to his approach have been proposed.

In (Hesselink and Roerdink, 2008), the authors introduced the Gamma $\gamma$-Integer Medial Axis (GIMA), where a point belongs to the skeleton if the distance between its two closest boundary points is at least equal to $\gamma$.
All the work mentioned above is based on the distance transform $\mathcal{D}_\Omega(x)$. The distance transform acts as a generator function for the medial axis, such that points $p \in \mathbf{MAT}$ if and only if they satisfy some constraint involving their distance to the boundary. However, by itself, the distance transform poses some problems because it usually contains discontinuities. See figure **4-2**. As a result, some studies focus on extracting alternative generator functions. An example is (Gorelick et al., 2006), where they proposed to estimating $\mathcal{D}_\Omega(x)$ as the solutions of the Poisson equation

$$\Delta u(x) = -1. \tag{4-1}$$

Later, (Aubert and Aujol, 2014) formalized the concept of the Poisson Skeleton and provided the details of a skeletonization algorithm based on this principle. Poisson skeletons rely on a solid mathematical formulation. Among other concepts, they use the local minimums and maximums of the curvature of $\delta\Omega$. However, when such methodology is applied in a discrete environment, many spurious branches appear due to the need to define the length of a kernel size to estimate these local extreme points.

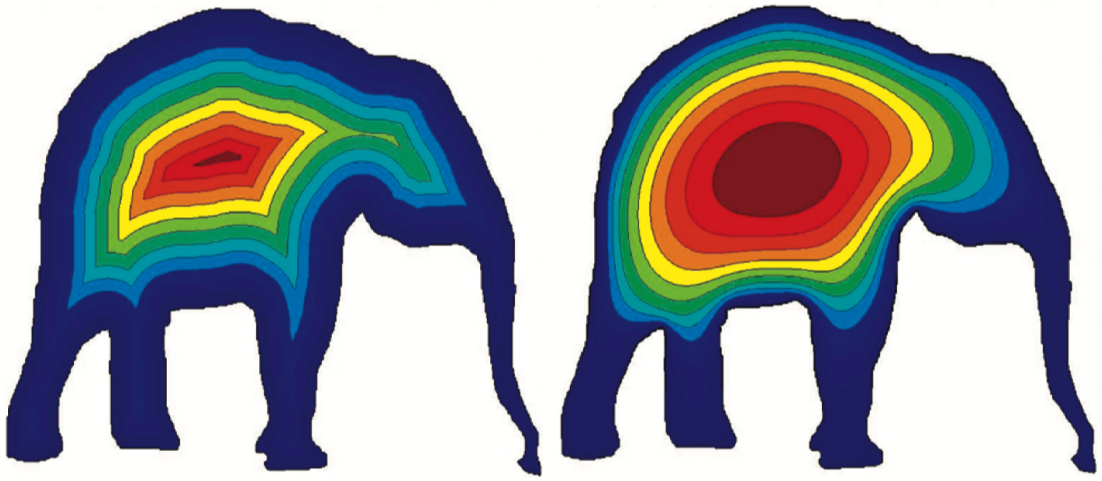


**Figure 4-2:** Distance transform discontinuities. The image in the left shows the Euclidean distance of each point to the contour of the object. Notice how some discontinuities appear near the areas where the level sets change curvature. The image on the right shows an at least twice differentiable approximation of $u$ using Gorelick et al. (2006) approach.

In contrast to the studies mentioned above, we propose a method to prune the medial axis and free it from as many spurious branches as possible without compromising the reconstruction

property. Our method must preserve the topology and connectivity of the original object while maintaining its equivariance to isometric transformations.

Let $\Omega$ be an n-dimensional closed shape with a boundary $\delta\Omega$. We propose a new pruning approach to robust skeletonization by filtering the **MAT** of $\Omega$ with a score function $\mathcal{F}_\Omega(X) : \mathbb{N}^2 \mapsto \mathbb{R}^+$. The function $\mathcal{F}_\Omega$ acts as an indication of the relevance of point $X$ in the **real** skeleton of $\Omega$. We define $\mathcal{F}_\Omega$ as the average of a set of estimation of the **MAT** over smoothed versions $\hat{\Omega}$ of the original object. Intuitively, $\mathcal{F}_\Omega$ acts as a sort of probability of how likely it is for a point $X$ to belong to the real skeleton of $\Omega$. The real skeleton branches will regularly appear in the skeletons resulting in high values of the score function. In contrast, spurious branches will only appear occasionally, resulting in low values.

To create the set of medial axis from the smoothed boundary, we used the Discrete Cosine Transform (DCT). The DCT in two dimensions has the form:

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{otherwise} \end{cases}$$

$C_v =$ (Similar to above)

$$\mathfrak{F}(u,v) = \frac{1}{4} C_u C_v \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \mathcal{I}(x,y) cos\left(u\pi \frac{2x+1}{2N}\right) cos\left(v\pi \frac{2y+1}{2M}\right)$$

Where $(u, v)$ are the frequency coordinates in the frequency domain.

The DCT is closely related to the discrete Fourier transform of real valued-functions. The DCT, however, has better energy compaction properties, with just a few of the transform coefficients representing the majority of the energy in the sequence. Multidimensional variants of the various DCT types follow straightforwardly from the one-dimensional definitions: they are simply a separable product (equivalently, a composition) of DCTs along each dimension. Mathematically, the DCT is perfectly reversible and does not lose any image information. To rebuild an image in the spatial domain from the frequencies obtained above, we use the IDCT:

$$\mathcal{I}(x,y) = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C_u C_v \mathfrak{F}(u,v) cos\left(u\pi \frac{2x+1}{2N}\right) cos\left(v\pi \frac{2y+1}{2N}\right). \tag{4-2}$$

For the n-dimensional DCT we will use the notation $\mathcal{I}(X)$, with $X \in \mathbb{R}^n$, instead of $\mathcal{I}(x, y)$. Additionally, we denote $\hat{\mathcal{I}}^{(M)}$ as the reconstructed version of $\mathcal{I}$ using only the first $M$ frequencies of the DCT in equation 4-2, with $M < N$. Now, we are ready to define our pruning strategy based on these concepts.

We define the **Cosine-Pruned Medial Axis** (CPMA) as a pruned version of the **MAT** of a shape $\Omega$. The **CPMA**($\Omega$) consist of all the pairs $(X, r) \in$ **MAT**($\Omega$) such that the score function evaluated on $X$ is greater than a threshold $\tau$. e.g. $\mathcal{F}_\Omega(X) > \tau$. The score function is defined as

$$\mathcal{F}_\Omega(X) = \frac{1}{M}\sum_{i=1}^{M}[\mathbf{MAT}(\hat{\mathcal{I}}^{(i)})](X). \tag{4-3}$$

The value of $\tau$ was determined empirically; however, we conducted an additional set of experiments to show how sensitive the CPMA is to different values of the threshold.
Although the CPMA results in a noise-free medial axis, its definition also allows disconnected skeletons. There is no restriction in its formulation to force individual elements of the CPMA to create a connected skeleton. We solved this issue by finding individual disconnected pieces of the CPMA and representing each piece as a graph. Later, we connect them using a geodesic distance $g(n_i, n_j)$ inside $\Omega$, where $n_i$ and $n_j$ where are nodes of two distinct pieces. However, the geodesic distance can lead to a connection between nodes that do not follow the medial axis (See figure **4-3**). To address this, we instead compute the minimum energy distance between $n_1$ and $n_2$ using the inverse score function as energy field, $E_\Omega = 1 - \mathcal{F}_\Omega$. Because $\mathcal{F}_\Omega$ is bounded on the interval $[0, 1]$, we guaranty that $E_\Omega(X)$ will have higher values as $X$ is close to $\delta\Omega$, and lower values when $X$ is close to the centerline of the object, hence forcing the paths to be close to the MAT. We call the result of connecting all the pieces the Connected CPMA.
In the next chapter, we will offer additional details about our method for robust skeletonization. We will present the implementation details of the concepts described above, as well as the algorithms and experiment results to support our claims.

### 4.2.2 Design of a Skeleton-Based Shape Descriptor

In this subsection, we describe the feature extraction mechanism to map the skeleton-based features discussed above to a set features $\phi$. We also describe the machine learning pipeline that takes $\phi$ as input and processes it to do both shape classification and retrieval. We will also show how the selected machine learning architecture preserves the invariant properties of the skeleton of the shape.
To extract a set of features from the skeleton representation of $\Omega$ through the use of the **Chordiogram** defined by Toshev (2011). The chordiogram is a rotation and translation invariant shape descriptor capturing global geometric relationships between elements of the object boundary. To define the chordiogram, consider a pair of boundary edges $p$ and $q$ from $\delta\Omega$. We will call such a pair $(p, q)$ a chord. One can define various features that describe the geometry of the chord, which we will denote by $f_{pq} \in \mathbb{R}^d$, as we will see in chapter 6. The chordiogram is defined as the k-dimensional histogram of $f_{pq}$ over all the chords. By

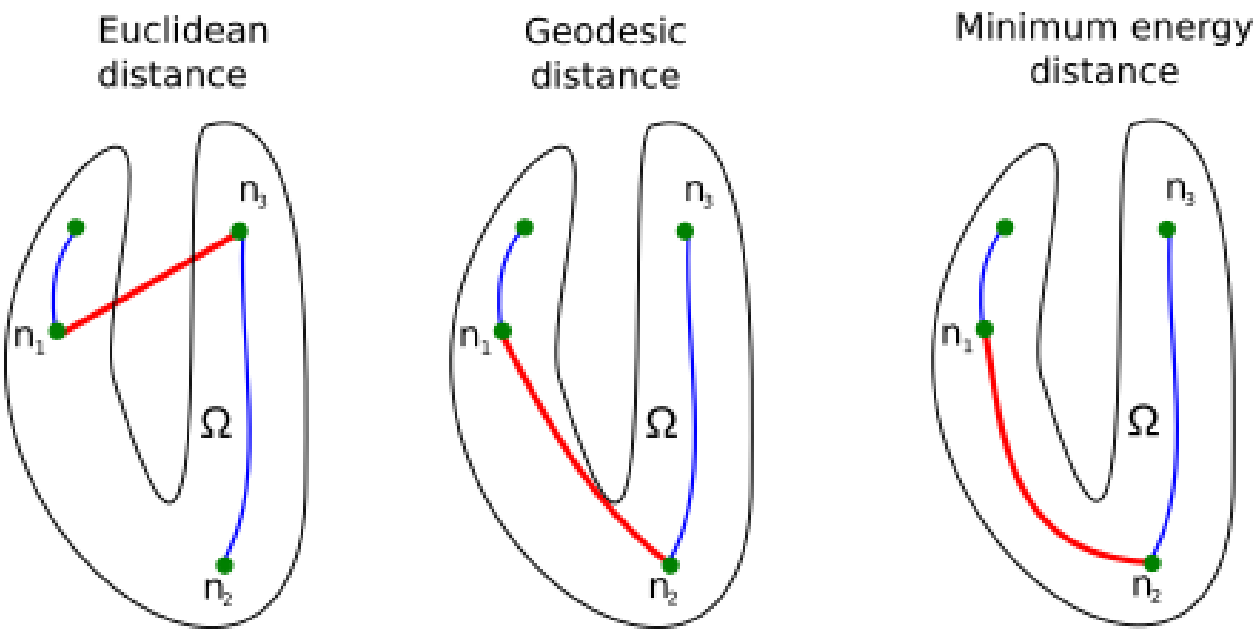


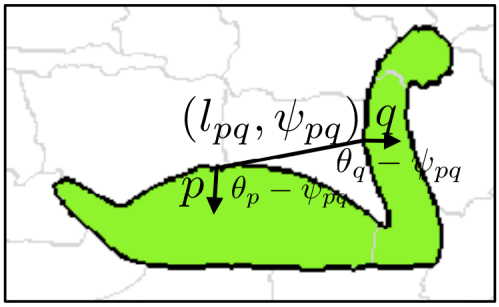

**Figure 4-3:** Path connectivity between CPMA segments. When using the euclidean distance (left), two nodes could connect through a path that is not entirely contained inside the object Ω. The minimum geodesic distance (center) guarantees that the path will be inside the object, but does not follow the centerline. The minimum energy distance with $\mathcal{F}_\Omega$ as the energy field is the best alternatives and forces the path to follow the medial axis.

carefully defining $f_{pq}$, and because isometric transformations are equally applied to all points in the object, the chordiogram is able to capture the invariant properties of Ω. e.g., if we set $f^0_{pq}$ as the euclidean distance between $p$ and $q$, and set $R$ as a rotation around the origin of coordinates; then we have that

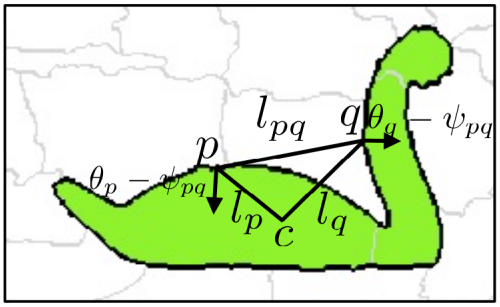
$$||Rp - Rq|| = ||R(p-q)|| = \sqrt{(p-q)^T R^T R(p-q)} = (p-q)^T(p-q) = ||p-q||, \qquad (4\text{-}4)$$

Because $R$ is a unitary matrix. Figure **4-4** shows how the chords are formed on a 2D shape. Two problems quickly arise from the definition of the chordiogram. First, the chordiogram depends on the relationships of pairs of points in the boundary, and then it needs a dense sampling to cover all the geometry of the object properly. Moreover, a dense sampling, such as $n$ points sampled from $\delta\Omega$, leads to a significantly large number of chords, $\#chords = n(n-1)$. This number quickly becomes intractable even for up-to-date computational resources, especially in the 3D case. With a small $n$, the risk is not to have enough representative points from the contour to capture its geometric properties.

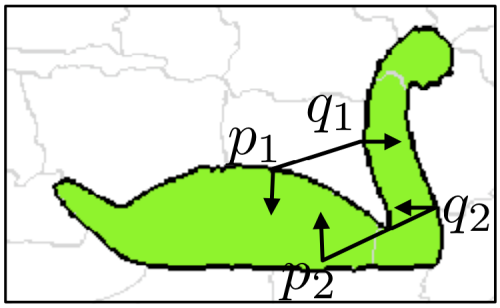
The second problem is the noise of the boundary. As in the case of the medial axis, boundary noise can cause variation of the final computation of the chordiogram. In this case, the noise is represented as jittering on the ends of the chords, slightly changing their features. Although




(a) Translation-invariant features.




(b) Rotation-invariant features.




(c) Normals.

**Figure 4-4:** Computation of individual chord in the chordiogram. Images a), b) c) shows examples of different chords from point pairs $(p_1, q_1)$ and $(p_2, q_2)$, an some of the properties that can extracted from them.

a good machine learning approach acting on a set of features should be robust to these types of artifacts in the data, this effect should be minimized.

To deal with sampling and boundary jittering problems, we will use results from subsection 4.2.1. After computing the CPMA of $\Omega$, we use it to extract chords from the joints of the skeleton, not from the boundary points. This is done to 1) reduce the total number of chords representing the object. Notice that the chords computed over the skeleton maintain the isometric invariances discussed in subsection 4.2.1.

Convolutional neural networks have shown that jointly optimizing feature extraction and classification pipelines can significantly improve object recognition (Lecun et al., 1998; Krizhevsky et al., 2012). Applying CNNs for shape analysis, however, is not as straightforward as one might think. CNN learning strategy is based on learning a set of features that are applied to the input as convolution filters. These mechanisms benefit from uniform grid structures of images in 2D, where convolutions can be computed via sliding windows; however, shape information is stored in different ways. In 2D, the segmented mask of a shape is a binary

image where there is not enough photometric variation in the image for the filters to learn useful features. Therefore, more preferable formats are used to obtain the shape information, e.g., a sample point cloud from the contour of the object. This strategy is especially useful in 3D, where information is typically represented as a point cloud or 3D triangular mesh.
For this reason, we chose to use PointNets (Qi et al., 2017a,b) as the machine learning model for our experiments. PointNets are deep learning architectures that directly consume raw point clouds without converting them to other formats. They have been applied to single object classification and semantic segmentation. Additionally, one of the key design aspects of PointNets is their invariance to permutations of the input points, which facilitates the learning process in datasets of objects coming from different sensors, resolutions of scales.
We combined our skeleton-based approach with the most recent and widely used architecture (PointNet++ arch (Qi et al., 2017b)). We did this by setting the list of $N$ chords computed from the skeleton of $\Omega$ as input for the neural network. The model determined by

$$\hat{Y} = \mathcal{L}_\theta(X) \tag{4-5}$$

Where $\theta$ is the parameter vector to train, and $X \in \mathbb{R}^{N\times K}$ is a list of $N$ chords each one with $K$ features. The network was trained using the classification cross-entropy as the loss function. After training, the learned latent space, the layer before the fully-connected part of the network, was used as a feature vector for retrieval.
In the next section, we will describe the experimental setup that was used to evaluate our proposed method.

## 4.3 Experimental setup

In this section, we explain in detail the experimental framework we used to evaluate the proposed descriptor. As we described in section 4.2, our approach to solve the stated problem has two main components: 1) a skeleton computation method that is robust to spurious branches; especially those coming from the noise in the contour of the object, and 2) the design of shape classification architecture based on features extracted from the topological skeleton. Thus, we designed a series of experiments to test each component of our solution in comparison to the state-of-the-art. We performed our experiments on available datasets of 2D silhouettes and 3D models in mesh format. Additionally, we employed well-known metrics to assess the main properties of our solution.

### 4.3.1 Datasets

We chose five extensively used datasets of 2D and 3D data to evaluate our methodology on skeletonization robustness, classification, and retrieval accuracy. These datasets are part of

the accepted benchmarks in literature, which enabled us to compare our results to previous work.

**Kimia216** In our comparative study, we used the Kimia-216 shape dataset (Sebastian et al., 2004) to test skeletonization robustness. It consists of 18 classes of different shapes with 12 samples in each class. The images in the dataset represent a collection of slightly different views of a set of shapes with different topologies. Contour noise and random rotations are also present in some of the images in the dataset. Kimia216 has been largely used to test a wide range of skeletonization algorithms. Because of this, and the large variety of shapes, we assumed that this benchmark ensured a fair comparison with the state-of-the-art regarding skeletonization methods. Figure **4-5** shows two samples from each class.

**Figure 4-5:** Kimia 216 dataset: Two samples shapes from each class.

**Animal Dataset** We used the Animal2000 (Bai et al., 2009) dataset to evaluate the performance of our skeleton representation with respect to articulations of the object; as a way to show the benefits of our proposed skeletonization method when facing non-rigid transformations. The Animal2000 database has 2000 images of 20 categories; each category consists of 100 images (Figure **4-6**). Because silhouettes in Animal2000 were obtained from real images, each class is characterized by a large intra-class variation in shape.

**MPEG7** We also experimented on the MPEG-7 CE-Shape 1 part B dataset (Latecki and Lakamper, 2000). This dataset is commonly used for the evaluation of shape-based classification and retrieval. It consists of 1400 binary object masks representing 70 different classes, each class including 20 examples. The metric employed to evaluate results on this dataset is the Bull's eye score: each shape is matched to all shapes, and the percentage of the 20 possible correct matches among the top 40 matches is recorded. The Bull's score is the average percentage of overall shapes. See Figure **4-7**.

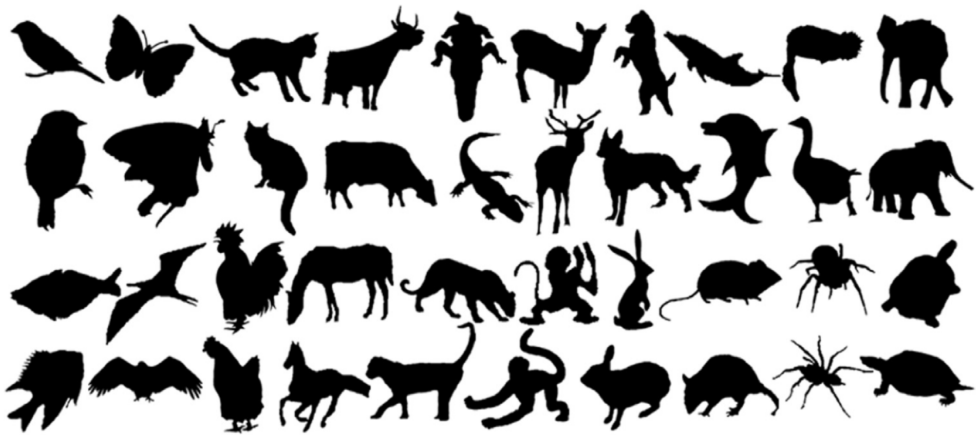


**Figure 4-6:** Sample shapes from Animal2000 database.

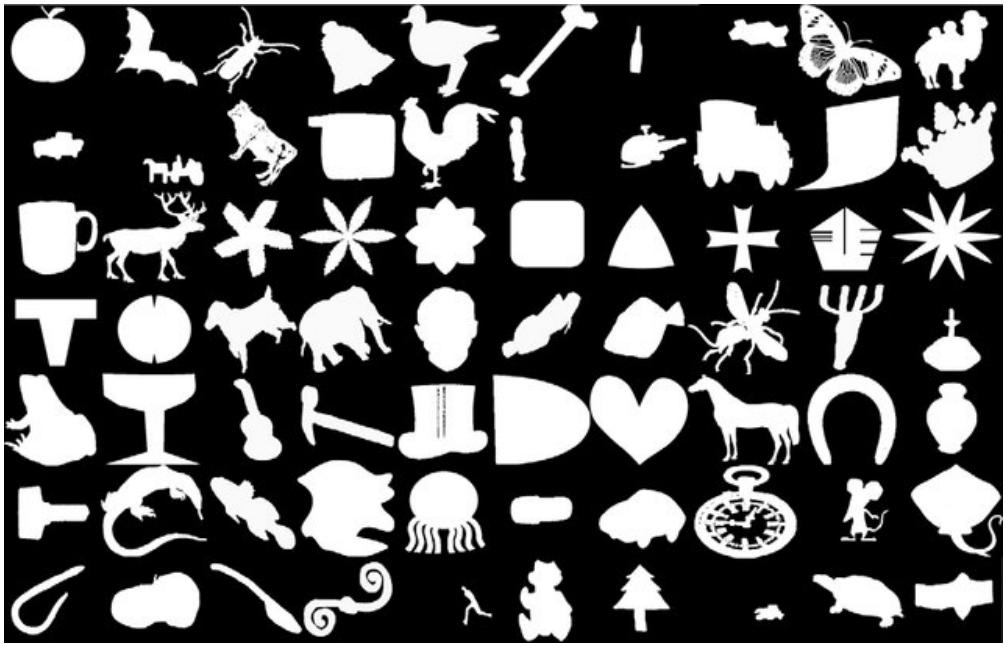


**Figure 4-7:** Examples from MPEG7 dataset: One sample from each category is shown.

**ModelNet** The main 3D dataset we used in this study is the well-known Princeton ModelNet-40 large-scale 3D CAD model dataset (Zhirong Wu et al., 2015). ModelNet-40 comprises 12.311 CAD models split into 40 categories. Additionally, the dataset splits into training and testing subsets containing 9843 and 2468 models, respectively. The models have been manually cleaned and normalized to fit into a unit sphere. ModelNet40 is often used as the benchmark for 3D shape recognition, shape classification and shape retrieval tasks (Qi et al., 2017a; Su et al., 2015); for this reason, it is suitable to evaluate the accuracy of the classification methodology developed in this dissertation. A rendered example from some categories can be referenced in Figure **4-8**.

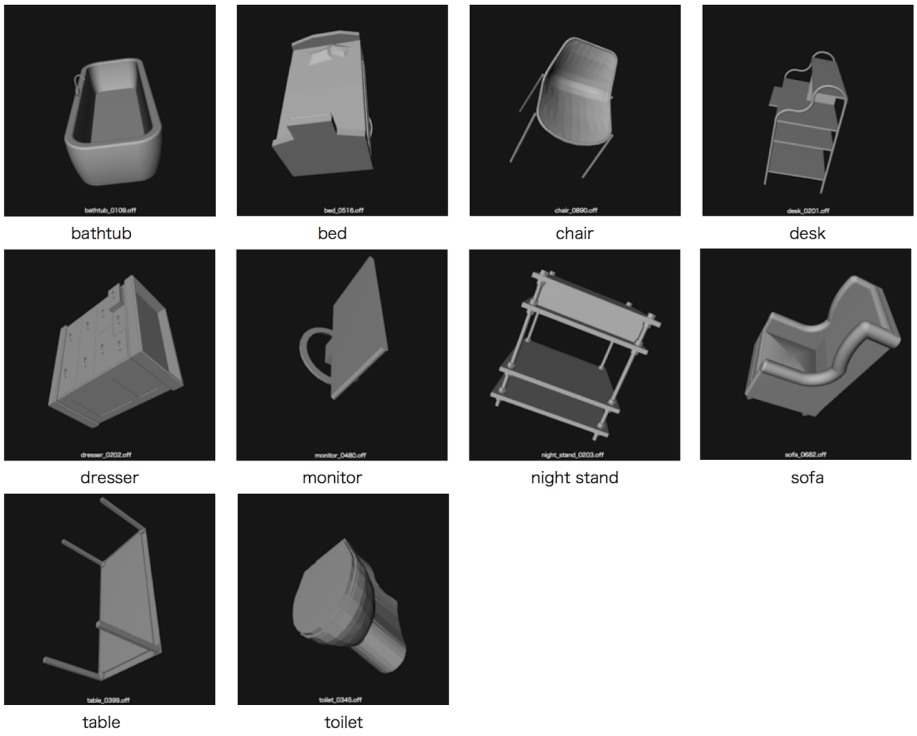



**Figure 4-8:** ModelNet-40 dataset: Rendered images from some of the 40 classes are shown in arbitrary poses.

**University of Groningen Skeletonization Benchmark** This set is commonly found in the literature to evaluate skeletonization methods in 3D (Sobiecki et al., 2014, 2013; Chaussard et al., 2011). It includes natural as well as synthetic shapes taken from other popular datasets. It includs shapes with and without (multiple) holes, of varying thickness, and with smooth and noisy boundaries. Shapes are provided in PLY triangle mesh format[1]. All meshes have been cleaned to ensure a consistent orientation, closeness, no duplicate or T vertices, and no degenerate faces. Each file describes a single connected shape. Mesh resolutions range between a few thousand vertices and over a million vertices. See Figure **4-9**.

### 4.3.2 Skeletonization Sensitivity Analysis

In order to compare the robustness of any skeletonization method, we adopted an evaluation strategy similar to the one presented in (Chaussard et al., 2011). Consequently, we measured the similarity between the medial axis of a shape $\Omega$ and the one of shape $\Omega'$ derived from a "perturbation" of $\Omega$. We were interested in evaluating how well our methodology would respond to the induced noise of the contour/surface of the object, which is a well-understood cause of spurious branches. We were also interested in assessing how stable the medial axis

[1] http://www.cs.rug.nl/svcg/Shapes/SkelBenchmark

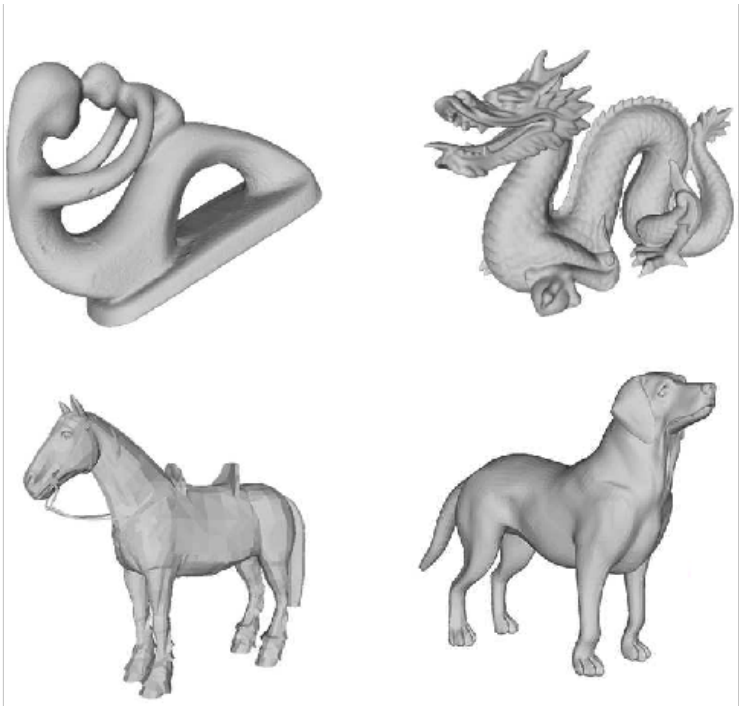


**Figure 4-9:** Groningen Skeletonization Benchmark: Examples of some of the in the dataset.

is in the presence of rotations to test for invariance to this isometric transformation.
As for similarity metrics, we employed the Hausdorff distance ($d_H$), and Dubuisson-Jain dissimilarity ($d_D$). The Dubuisson-Jain similarity is essentially a normalization of the Hausdorff distance using the size of the set involved in the computation (Dubuisson and Jain, 2002), to help to overcome $d_H$ sensitivity to outliers. The Dubuisson-Jain similarity is defined as

$$d_D(X,Y) = max\left\{D(X|Y), D(Y|X)\right\}, \tag{4-6}$$

with

$$D(X|Y) = \frac{1}{|X|}\sum_{x\in X}\min_{y\in Y}\left\{d(x,y)\right\}. \tag{4-7}$$

To evaluate the noise sensitivity, we must first choose a strategy to induce noise to the boundary of the object. We used a stochastic approach where a point $p$ and its neighbors $N(p)$ are deformed by a vector $v$ in the direction orthogonal to the boundary, and a deformation magnitude normally distributed, $|v| \sim \mathcal{N}(\mu = p, \sigma = 1)$.
This noise model is recurrently applied $n$ time to every shape in our datasets. We denote as $\mathbf{MAT}_i(\Omega)$ the medial axis of a shape $\Omega$ after applying the noise model $i$ times. To determine how invariant a particular skeletonization method is to boundary noise, we compared the original medial axis $\mathbf{MAT}(\Omega)$ with the noisy version of it $\mathbf{MAT}_i(\Omega)$ using both similarity

metrics. We use similarity over all the elements in the dataset as the measurement of the invariance of the skeletonization method.

Regarding rotation sensitivity, the medial axis is ideally a rotation-invariant shape descriptor so that $\mathrm{MA}(R(\Omega)) = R(\mathrm{MA}(\Omega))$. See equation 2-2. Due to sampling factors, this relationship is merely an approximation. However, we can measure how "invariant" a medial axis is by comparing $\mathbf{MAT}(R(\Omega))$ with $R(\mathbf{MAT}(\Omega))$ for different objects, and different definitions of the medial axis. The more similar they are, on average, the more invariant the skeletonization method is at computing the medial axis.

Rotations range from 0 degrees to 90 degrees counterclockwise around the origin for 2D objects in our experiments. In 3D, we used a combination of azimuthal ($\theta = [0, \frac{\pi}{2}]$) and elevation ($\phi = [0, \frac{\pi}{2}]$) rotations around the origin, so that the final rotation matrix is:

$$R = R_{az}(\theta)R_{el}(\phi) \tag{4-8}$$

Both the noise sensitivity and the rotation sensitivity tests were conducted on Kimia's and Animal2000 datasets in 2D. For the 3D case, we usesd the Groningen Skeletonization Benchmark.

### 4.3.3 Shape Classification and Retrieval

One of the goals of this dissertation is to tackle the problem of invariance to isometric transformation. We, therefore, focused our experiments on problems that benefit from such invariance, namely, shape classification and retrieval in arbitrary orientations. We conducted shape classification and shape retrieval experiments on the MPEG7 and ModelNet-40 datasets to show the performance of our methodology.

The per-instance accuracy is the most common metric used to assess the performance of the classification. By using accuracy, we were able to compare our results with state-of-the-art methods. These methods will be discussed in detail in the next chapters. For the classification, we used ModelNet-40 due to a well-established classification benchmark associated with this dataset. Unlike the benchmark, we considered three experimental modes to highlight the advantage of our classification pipeline most effectively. The three modes include (1) training and testing with azimuthal rotations (z/z), (2) training and testing with arbitrary rotations ($\mathbf{SO(3)}/\mathbf{SO(3)}$), and (3) training with azimuthal and testing with arbitrary rotations (z/$\mathbf{SO(3)}$).

We used the MPEG7 dataset for our retrieval experiments. In literature, the retrieval rate for this dataset is measured by the so-called Bull's Eye score. Every shape in the database is compared to all other shapes, and the number of shapes from the same class among the 40 most similar ones is reported. The Bull's Eye retrieval rate is the ratio of the total number of shapes from the same class to the highest possible number (which is 20 x 1400). Thus,

the best possible rate is 1:

$$BS = \frac{D_{40}}{P}. \tag{4-9}$$

Where $D_{40}$ is the total sum of correct retrieval instances out of the 40 most similar objects, and $P = 20 \times 1400$ is the total possible outcome.
Results of the $BS$ computed on ModelNet40 and classification accuracy on MPEG7 dataset will also be reported and discussed in the following chapters. However, because the benchmarks of the datasets do not use these alternative metrics, it is not possible to show comparative results.

## 4.4 Limitations

The goal of this research is to study skeleton-based shape description in detail. We aim to explore the properties that make topological skeletons adequate to approach the problem of isometric-invariant shape representation. Despite the long list of classification and retrieval approaches that use a wide range of features inherent to the objects, we are only interested in their shape. Therefore, no photometric or texture information was used in this study. Consequently, the selected datasets only contain information about the geometry of the objects in the form of pre-segmented silhouettes in the 2D case, or triangular meshes without texture in the case of 3D models.
Additionally, in this study, we are interested in exploring isometric invariances and equivariances in shape analysis. For that purpose, we create a classification pipeline based on a deep learning model. Because we do not aim to explore the best deep architecture to classify shapes, we set PointNet++ as our machine learning model. However, we acknowledge that other approaches exist and should be fairly compared with our results.
Although in chapter 1, we discussed a few applications fields where shape analysis has an impact; we do not aim to present any applications of the research problem of this dissertation. This work is limited to assess the performance of the new proposed descriptor in comparison to well-established datasets and benchmarks in the scientific literature.
In this chapter, we defined the employed methodology for this research by describing the two key elements of this work: 1) robust skeletonization and 2) design of classification pipeline based on topological features. In the next chapters, we will further detail both elements and provide a detailed explanation of the experiments conducted and their results.

# 5 Robust Skeletonization

In this chapter, we address the problem of spurious branch removal. Spurious branches in the MAT are associated with noise in the contour. Even a small perturbation can cause a new branch to appear. To address this challenge, we developed a method that produces a consistent skeleton in the presence of several degrees of noise, and show the results of the experiments designed to prove our robust skeletonization. We conducted additional experiments to demonstrate how our method, the CPMA, exhibits equivariance to isometric transformations.

## 5.1 The Cosine-Pruned Medial Axis

The **Cosine-Pruned Medial Axis** is defined as a pruned version of the **MAT** of a shape $\Omega$. The computation of the CPMA starts with the estimation of the Discrete Cosine Transform ($DCT(\Omega)$). We can reconstruct the original object from the DCT using a truncated number of frequencies $M$. We repeat this process several times, increasing the value of $i$ each time. By using this method, we assure that each reconstruction is a smooth version of the original object. Each smooth reconstruction is expected to have a skeleton with less spurious branches than the original while still maintaining the skeletal structure.
After computing a set of reconstructions $\mathcal{S} = \left\{\mathcal{I}^{(i)}\right\}$ for $i = 1, 2, ..., M$; we aggregate all of them to create a score function using equation 4-3, and denote it $\mathcal{F}_\Omega$ . Later, the CPMA is the result of thresholding $\mathcal{F}_\Omega$ with the parameter $\tau$.
From the above description, it is possible to see how the CPMA can result in a disconnected skeleton. Despite the fact that the CPMA is invariant to contour noise, the connectivity property is necessary to ensure that the topology of the original object is preserved. For this reason, we enforce such connectivity by iteratively connecting the individual parts of the CPMA through a minimum energy path approach. We will refer to this result as the CPMA + connectivity.
In the following sections, we summarize the implementation issues encountered while implementing our methodology. We also offer proof of the isometric equivariance of the CPMA.

### 5.1.1 Implementation details

In this section, we discuss the algorithms that we used to implement the CPMA and the CPMA + connectivity efficiently. We also discuss their complexity and hyper-parameter

selection.

We based our code in python programming language due to its simplicity, the vast number of libraries available, and the popularity in the scientific community for academic projects, particularly in machine learning.

The CPMA only relies on one parameter, $\tau$. The value of $\tau$ is a threshold to determine whether a point of $\mathcal{F}_\Omega$ is a skeleton point. The value of $\tau$ was defined empirically to be $\tau = 0.5$; however, we conducted an additional experiment to show how sensitive the CPMA is to different values of the threshold. This experiment will be described later in subsection 5.2.4.

Another important consideration to take into account when computing the CPMA is the number of frequencies employed on the reconstruction of the original object through the DCT. We found that using frequencies greater than $\frac{N}{2}$ does not yield significant improvement for the CPMA. Here, all of the objects are enclosed in square images or cubic volumes with side $N$ in length.

Finally, we describe the implementation issues of connectivity enforcement. We connect the different pieces of the raw CPMA through the minimum energy path, as stated in chapter 4. To do so, we create a lattice graph $\mathcal{G} \in \mathbb{Z}^n$. A point $p$ representing a pixel or a voxel, is a node of $\mathcal{G}$, if and only if $p \in \Omega$. The node $p$ shares an edge with every one of its neighbors in the lattice if they are inside $\Omega$. We used an 8-connectivity neighborhood in 2D and a 26-connectivity neighborhood in 3D.

To determine the minimum energy path between pairs of pixels/voxels with this methodology, we compute the minimum path in the graph using Dijkstra's algorithm. The weights of each edge are the average of two neighboring values $\mathcal{E}_\Omega$, e.g.

$$\text{weight}(e_{x,y}) = 0.5(\mathcal{E}(x) + \mathcal{E}(y)).$$

This method guarantees the connectivity, but it is inefficient because it is based on an iterative computation of the minimum energy path. We sacrifice performance in favor of connectivity. The algorithm for the computation of the CPMA is presented in detail in algorithm 1. Additionally, the connectivity enforcement procedure is explained in algorithm 2.

### 5.1.2 Isometric Equivariance of the CPMA

Because distance transform-based skeletons depend only on the shape $\Omega$, and not on the position or size in the embedding space, skeletons should be equivariant under isometric transformations $R$ of $\Omega$, i.e., $\mathbf{MAT}(R(\Omega)) = R(\mathbf{MAT}(\Omega))$. To prove this equivariance, we start from the definition of the MAT

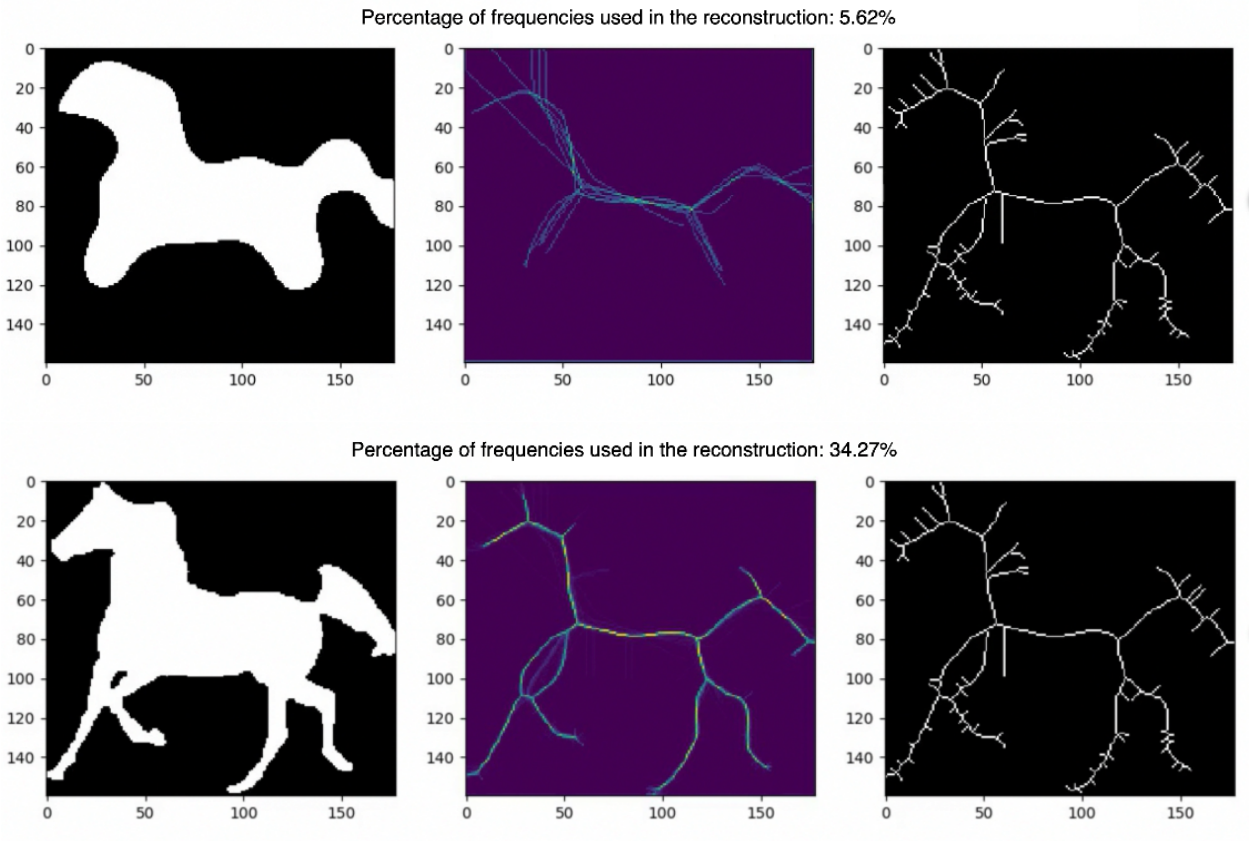



**Figure 5-1:** Score Function illustrative example. The images show from left to right the reconstructed object $\Omega$, the score function $F_\Omega$ and the real medial axis of $\Omega$. The first row shows the $F_\Omega$ computed with reconstructions up to only $M_1$ of the first frequencies. The second row shows $F_\Omega$ computed with reconstructions up to $M_2$ of the first frequencies. $M_2 > M_1$.

$$\mathbf{MAT}(\Omega) = \{(X, r) \mid X \in \Omega \wedge B_r(X) \nsubseteq B_{r'}(X'), \forall r' > r, X' \in \Omega\},$$

where $B_r(X)$ is closed ball centered in $X$ with radius $r$. If we apply an isometric transformation $R$ on $\mathbf{MAT}(\Omega)$ we obtain

$$R(\mathbf{MAT}(\Omega)) = \{(R(X), r) \mid X \in \Omega \wedge B_r(X) \nsubseteq B_{r'}(X'), \forall r' > r, X' \in \Omega\}.$$

To compute $R(\mathbf{MAT}(\Omega))$ we only need to transform the elements of the result set. As the transformation does not have any effect on the radius, it is easy to verify that $X \in \Omega$ if and only if $R(X) \in R(\Omega)$. As a consequence, $B_r(R(X)) \nsubseteq B_{r'}(R(X')), \forall r' > r, R(X') \in R(\Omega)$. Let us now define $Y = R(X)$. We then have that

**Algorithm 1:** Cosine-Pruned Medial Axis (CPMA)

**Input:**

$\mathcal{I}$: N-dimensional binary array representing the object
**M:** number of frequencies of $\mathcal{I}$ to be used in the computation

**Output:**

**CPMA:** Cosine-Pruned Medial Axis

$\tau \leftarrow 0.5$
$\mathfrak{F} \leftarrow DCT(\mathcal{I})$
$i \leftarrow 1$
**while** $i < M$ **do**
  $\hat{\mathcal{I}}^{(i)} = IDCT(\mathfrak{F}, i)$  // Reconstruction of $\mathcal{I}$ using only the first $i$ frequencies
  $\mathcal{F}_\Omega = \mathcal{F}_\Omega + \mathbf{MAT}(\hat{\mathcal{I}}^{(i)})$
  $i \leftarrow i + 1$
**end**
$\mathcal{F}_\Omega \leftarrow \mathcal{F}_\Omega / M$  // The final $\mathcal{F}_\Omega$ is the average of all reconstructions

CPMA $= \mathcal{F}_\Omega > \tau$
**return** CPMA

$$
\begin{aligned}
R(\mathbf{MAT}(\Omega)) &= \left\{(R(X), r) \;\middle|\; X \in \Omega \;\wedge\; [B_r(X) \nsubseteq B_{r'}(X'), \forall r' > r, X' \in \Omega]\right\} \\
&= \left\{(R(X), r) \;\middle|\; R(X) \in R(\Omega) \;\wedge\; [B_r(R(X)) \nsubseteq B_{r'}(R(X')), \forall r' > r, R(X') \in R(\Omega)]\right\} \\
&= \left\{Y, r) \;\middle|\; Y \in R(\Omega) \;\wedge\; [B_r(Y) \nsubseteq B_{r'}(Y'), \forall r' > r, Y' \in R(\Omega)]\right\} \\
&= \mathbf{MAT}(R(\Omega))
\end{aligned}
$$

Our method, the CPMA, depends entirely on $\mathcal{F}_\Omega$, which also holds the isometric equivariant property. In fact, by using the above result, and recalling that $R$ is a linear transformation, we can demonstrate that

$$
R(\mathcal{F}_\Omega) = R\left(\frac{1}{M}\sum_{i=1}^{M}\mathbf{MAT}(\hat{\mathcal{I}}^{(i)})\right) = \frac{1}{M}\sum_{i=1}^{M} R\left(\mathbf{MAT}(\hat{\mathcal{I}}^{(i)})\right) = \frac{1}{M}\sum_{i=1}^{M}\mathbf{MAT}\left(R(\hat{\mathcal{I}}^{(i)})\right) = \mathcal{F}_{R(\Omega)},
$$

hence proving that the CPMA is equivariant to isometries.
Analytic methods typically satisfy this property because all computations are done in high-precision, continuous vector space. In contrast, discrete methods cannot be fully equivariant because samples of both $\Omega$ and MAT($\Omega$) are constrained to the fixed voxel grid. Additionally, the equivariance of the MAT is also affected by the use of the discrete cosine transform. In a continuous domain, it is easy to demonstrate that the cosine transform, as well as the Fourier Transform, have exact

**Algorithm 2:** Connect skeleton segments

**Input:**
**CPMA:** Cosine-Pruned Medial Axis
representing the object
$\mathcal{F}_\Omega$ Score function
**Output:**
**C-CPMA:** Connected Cosine-Pruned Medial Axis

```
C-CPMA ← copy(CPMA)
skeleton-parts ← compute-skeleton-parts(CPMA)
max-iter ← 200
it ← 0
while it < max-iter and |skeleton-parts| > 1 do
    graph-i ← skeleton-parts[0]
    graph-f ← skeleton-parts[1]
    // Finds the minimum path inside the object for two pieces of the skeleton
    min-path ← find-min-path(graph-i, graph-f, F_Ω)
    C-CPMA[path] ← True
    skeleton-parts ← compute-skleton-parts(C-CPMA)
    it ← it + 1
end
return C-CPMA
```

isometric equivariance; however, in a discrete domain, this equivariance is only an approximation, $R(\mathcal{F}_\Omega) \approx \mathcal{F}_{R(\Omega)}$.

## 5.2 Experiments and Results

### 5.2.1 Comparative Studies

We chose a set of seven of the most relevant methods in the scientific literature to compare with CPMA skeletonization results. These methods were selected based on a careful review of the state-of-the-art on skeletonization. These methods were also chosen to illustrate the variety of approaches authors employ to pruning the medial axis. The first method we used in our comparative study is the MAT itself without any pruning. The MAT is computed by first calculating the distance transform of the image. The MAT then lies along the singularities (i.e., creases or curvature discontinuities) in the distance transform.

We chose the most representative pruning methods of the state-of-the-art. In general, pruning methods use parameters derived from the distance transform and the projection of a point $x$ in the MAT to the nearest background point of the object, $P_x$. However, other factors, such as contour curvature, are also used. Table **5-1** summarizes all of the studies employed in our comparative study for 2D. For each method, we used values for the parameters that are the most common

through other comparative studies in previous works. In the following sections, we will also show how the CPMA parameter $\tau$ is stable across different objects and even different datasets.

| Method | Abbreviation | Authors | Parameter |
|---|---|---|---|
| 2D methods | | | |
| Medial Axis Transform | MAT | Blum (1967) | N/A |
| Zhang-Suen Algorithm | Thinning | Zhang and Suen (1984) | N/A |
| Gamma Integer Medial Axis | GIMA | Hesselink and Roerdink (2008) | $\gamma$: minimum distance between $\mathcal{P}_x$ and $\mathcal{P}_y$, $y \in N_x$. |
| Bisector Euclidean Medial Axis | BEMA | Couprie et al. (2007) | $\theta$: angle formed by the point $x$ and the two projections $\mathcal{P}_x$ and $\mathcal{P}_y$, $y \in N_x$. |
| Scale Axis Transform | SAT | Giesen et al. (2009) | **s**: scale factor to resize $\mathbf{MAT}(\Omega)$. |
| Scale Filtered Euclidean Medial Axis | SFEMA | Postolski et al. (2014) | **s**: scale factor for individual balls in the $\mathbf{MAT}(\Omega)$. |
| Poisson Skeleton | Poisson skel. | Aubert and Aujol (2014) | **w**: window size to find the local maximum of contour curvature. |
| 3D methods | | | |
| Zhang-Suen Algorithm | Thininng | (Zhang and Suen, 1984) | N/A |
| Tree-structure skeleton extraction | TEASAR | (Sato et al., 2000) | N/A |

**Table 5-1:** Pruning methods employed for the comparative study in 2D. The table shows author, name, and parameter description for each method. The point $x \in \mathbf{MAT}$ is element of the MAT that might be pruned. $P_x$ refers to the closest boundary point of $x$, while $N_x$ accounts for the neighborhood around it.

### 5.2.2 Stability Under Noisy Boundary

Medial axes are notoriously sensitive to border noise. Because we argue that the CPMA cope reasonably well with shape deformation, it is useful to test how it performs in practice. We, therefore, compared the stability of the CPMA with the other approaches in table **5-1**, with respect to contour noise. We based our 2D experiments on Kimia216 and the Animal Dataset for 2D. Additionally, we used a set of three-dimensional objects from the Groningen Benchmark for 3D.
To introduce noise to the boundary of an object, we use two methods: one for 2D and one for 3D. In the 2D case, we deform the random points in the contour of the object in a direction orthogonal

to them. Let $\gamma(i) = [x(i), y(i)]$ be the coordinates of the point $i$ in the contour, and let $N_i$ be all the neighbor points of $\gamma(i)$ in the contour. The deformation process is applied to a random number of points in the contour such that

$$\hat{\gamma}(j) = \gamma(j) + \delta.\lambda.\vec{v} \ , \ \forall j \in N_i \tag{5-1}$$

Where $\vec{v} = [\ -y'(i) \quad x'(i)\ ]^T$ is a vector normal to the curve in $i$, $\delta$ randomly takes the values 1 or $-1$, and $\lambda$ is a random variable such that $\lambda \sim \mathcal{N}(i, \sigma)$. Our noise model produces protrusions around random points in the contour chosen with uniform distribution and probability of be deformed of $p = 0.005$.

In the 3D case, we propose a simple mechanism to deform the object using a process derived from the Eden's (accretion) process. The Eden's process is an iterative random cellular automaton that, in its simplest form, attributes an equal probability to all the outer border points to be set to 1 at each step. That is, at each step, a neighbor of the object is chosen randomly and added to the shape. As a result, the object's homotopy type remains unchanged at each step. For both 2D and 3D, we denote by $E(\Omega, k)$ the result of applying $k$ steps of the noise model to the shape $\Omega$.

For our noise sensitivity experiments, we apply 20 times $E(\Omega, k)$ to every object of every dataset, and then compute its MAT using every method in table **5-1** with different parameters. Later, each $\mathbf{MAT}(E(\Omega, k))$ is compared with $\mathbf{MAT}(\Omega)$ using both the Hausdorff distance and Dubuisson-Jain dissimilarity. Finally, we report the per method average of each metric over all the elements of each dataset.

We first test our skeletonization method on the Kimia216 dataset. We used all methods from table **5-1** with different input parameters, and compared them to the CPMA and the CPMA with connectivity enforcement. The parameters for each method were chosen empirically as the most commonly employed parameters in literature. The results for all methods are shown in table **5-2**. This table shows how our methods are competitive against state-of-the-art skeletonization methods such as the GIMA and SFEMA, and perform better than methods such as Poisson Skeletons, SAT, skeletonization by thinning, and the MAT itself. Figure **5-2** shows both Hausdorff distance and Dubuisson-Jain dissimilarity against noise level. The figure only displays the curves with the parameters that yielded the best performance for every method in the comparative study. Note that for most methods, the results are approximately the same for low levels of noise; however, the curves (and the values in the table) start diverging when the noise level increases, revealing the real noise invariant properties. In the figure, it is also possible to see how the CPMA and the CPMA+connnect are among the three best results when the dissimilarity metric is used, only surpassed by the GIMA. Although the results are not as good compared to methods with the Hausdorff distance, it is possible to see how our methods still fall among the top five results.

The Animal2000 dataset contains nearly ten times more shapes than Kimia216. The Animal2000 dataset consists of 2000 images of different animals in a wide variety of poses. This implies more variation between shapes, and therefore a more challenging setting. Table **5-3** shows similar results compared to Kimia216, confirming that the noise invariant properties of the CPMA still hold in a more robust dataset. The GIMA is still the best method measured with the Dubuisson-Jain dissimilarity, and it is followed closely by both the CPMA and the CPMA with connectivity

| **Method** | **Hausdorff** | | | | **Dissimilarity** | | | |
|---|---|---|---|---|---|---|---|---|
| | 5 | 10 | 15 | 20 | 5 | 10 | 15 | 20 |
| MAT | 8.13 | 8.50 | 8.43 | 9.41 | 1.95 | 2.67 | 3.01 | 3.27 |
| Thinning | 4.68 | 5.85 | 6.88 | 8.15 | 2.18 | 3.26 | 3.94 | 4.45 |
| GIMA (r=5) | 5.46 | 6.50 | 7.37 | 8.84 | 0.87 | 1.31 | 1.60 | 1.88 |
| GIMA (r=10) | 5.40 | 7.12 | 8.35 | 9.18 | 0.68 | 1.08 | 1.35 | 1.58 |
| **GIMA (r=20)** | **4.49** | **5.76** | **6.39** | **7.30** | **1.00** | **1.30** | **1.05** | **1.35** |
| BEMA (theta=90) | 5.22 | 6.55 | 7.11 | 8.30 | 0.99 | 1.60 | 2.07 | 2.53 |
| BEMA (theta=120) | 5.05 | 6.56 | 7.60 | 8.74 | 0.70 | 1.37 | 1.94 | 2.52 |
| BEMA (theta=150) | 6.68 | 7.69 | 7.89 | 9.40 | 0.99 | 1.80 | 2.50 | 3.37 |
| SAT (s=1.1) | 8.68 | 8.73 | 8.76 | 9.64 | 3.09 | 4.37 | 5.08 | 5.57 |
| SAT (s=1.2) | 9.61 | 10.05 | 9.79 | 10.20 | 2.50 | 3.22 | 3.89 | 4.47 |
| SFEMA (s=1.1) | 4.15 | 5.35 | 6.18 | 7.53 | 0.84 | 1.37 | 1.92 | 2.50 |
| SFEMA (s=1.2) | 3.64 | 5.13 | 6.15 | 7.64 | 0.68 | 1.11 | 1.53 | 1.99 |
| Poisson skel. (w=0.05) | 11.43 | 11.16 | 11.26 | 12.73 | 2.46 | 3.05 | 3.27 | 3.53 |
| Poisson skel. (w=0.10) | 15.60 | 15.48 | 16.07 | 17.35 | 3.62 | 4.07 | 4.19 | 4.56 |
| **Poisson skel. (w=0.20)** | **17.71** | **18.02** | **19.54** | **21.35** | **5.00** | **5.38** | **5.63** | **6.20** |
| CPMA | 5.55 | 7.28 | 8.07 | 9.66 | 0.71 | 1.07 | 1.39 | 1.71 |
| CPMA + Connect. | 5.19 | 6.68 | 7.66 | 9.12 | 0.80 | 1.20 | 1.58 | 1.94 |

**Table 5-2:** Noise sensitivity results on Kimia216. The table shows the average Hausdorff distance and Dubuisson-Jain dissimilarity for different noise levels (5-20) over each element of the dataset. The best and worst method are highlighted to facilitate the comparison with the CPMA.

enforcement. Results of using the Hausdorff distance as a metric show that the CPMA is close to methods such as BEMA and SFEMA; however, the results are not as good as when using the dissimilarity. Figure **5-3** depicts the best performance for every method in comparison to ours. Our results suggest that the CPMA noise invariant properties generalize across different datasets because of the complexity of Animal2000 regarding the number of shapes and the diversity of those shapes. Note that that results in figure **5-3** appear to be smoother than those in the previously discussed figure **5-2** on Kimia216. For further inspection, we include figures of all methods and all parameters in appendix **??**.

For our 3D experiments, we picked 14 objects from the Groningen benchmark, reflecting the most common shapes used in the literature. Each object was voxelized to a binary voxel grid with resolution $150 \times 150 \times 150$. This resolution offers sufficient details as well as a sufficiently low computational cost. In contrast to the 2D case, we apply $E(\Omega, k)$ only 10 times to the 3D object. We did this for two reasons: 1) to reduce computational complexity, and 2) because in 3D with

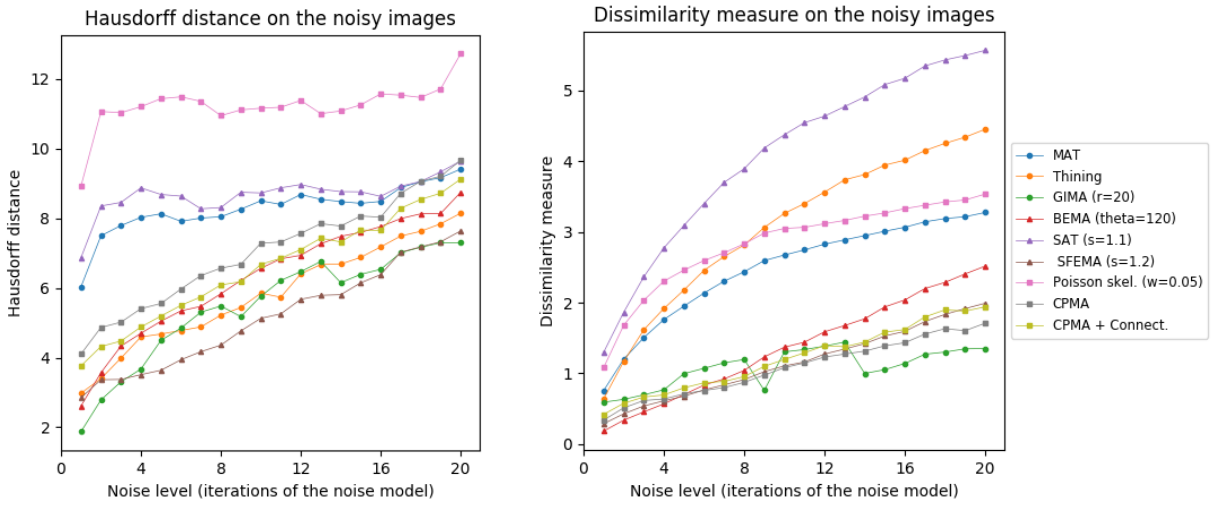



**Figure 5-2:** Noise sensitivity results on Kimia216 dataset. The figure shows the Hausdorff distance (left) and the Dubuisson-Jain dissimilarity (right) for all the methods in table **5-2**. Only the best parametrization of each method is depicted for better interpretation.

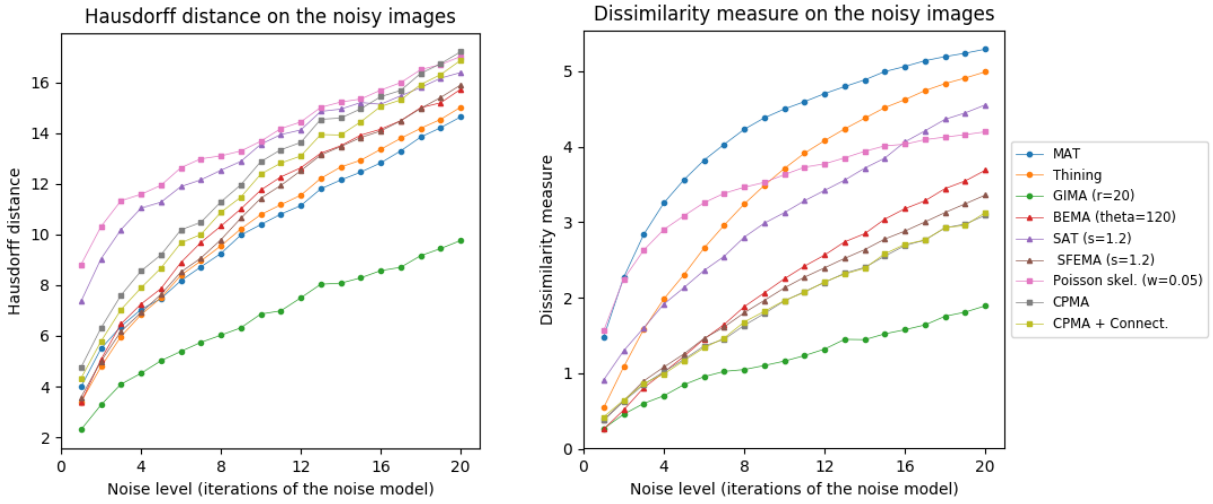



**Figure 5-3:** Noise sensitivity results on Animal2000 dataset. The figure shows the Hausdorff distance (left) and the Dubuisson-Jain dissimilarity (right) for all the methods in table **5-3**. Only the best parametrization of each method is depicted for better interpretation.

the chosen resolution, noise tends to be more extreme. The results on the Groningen dataset are shown in table **5-4** and figure **5-4**. We can observe that both the CPMA and CPMA+connectivity achieved the best results among other methods when compared with the dissimilarity measure. These results are evidence that our methodology has noise-invariance properties, and it is stable in the presence of contour of surface deformation. However, the results show unusual patterns when

| Method | Hausdorff | | | | Dissimilarity | | | |
|---|---|---|---|---|---|---|---|---|
| | 5 | 10 | 15 | 20 | 5 | 10 | 15 | 20 |
| MAT | 7.47 | 10.39 | 12.47 | 14.64 | 3.56 | 4.50 | 5.00 | 5.29 |
| Thinning | 7.51 | 10.79 | 12.94 | 15.03 | 2.30 | 3.71 | 4.52 | 4.99 |
| GIMA (r=5) | 7.96 | 11.00 | 13.23 | 15.27 | 1.24 | 1.88 | 2.35 | 2.68 |
| GIMA (r=10) | 6.78 | 8.56 | 10.21 | 11.54 | 0.89 | 1.29 | 1.62 | 1.89 |
| **GIMA (r=20)** | **5.02** | **6.86** | **8.29** | **9.77** | **0.85** | **1.16** | **1.52** | **1.89** |
| BEMA (theta=90) | 7.84 | 10.61 | 12.76 | 14.78 | 1.45 | 2.37 | 3.06 | 3.57 |
| BEMA (theta=120) | 7.86 | 11.76 | 13.93 | 15.74 | 1.22 | 2.25 | 3.04 | 3.69 |
| BEMA (theta=150) | 8.88 | 12.38 | 14.39 | 16.51 | 1.68 | 3.00 | 3.95 | 4.72 |
| SAT (s=1.1) | 9.44 | 11.80 | 13.47 | 15.21 | 2.80 | 4.13 | 5.02 | 5.49 |
| SAT (s=1.2) | 11.28 | 13.57 | 15.21 | 16.40 | 2.13 | 3.13 | 3.85 | 4.55 |
| SFEMA (s=1.1) | 7.44 | 11.00 | 13.30 | 15.35 | 1.33 | 2.27 | 3.03 | 3.69 |
| SFEMA (s=1.2) | 7.64 | 11.43 | 13.83 | 15.90 | 1.26 | 2.13 | 2.78 | 3.36 |
| Poisson skel. (w=0.05) | 11.94 | 13.68 | 15.35 | 17.03 | 3.08 | 3.63 | 4.01 | 4.20 |
| Poisson skel. (w=0.10) | 14.55 | 17.11 | 18.64 | 20.10 | 3.55 | 4.08 | 4.68 | 4.94 |
| **Poisson skel. (w=0.20)** | **17.37** | **20.35** | **21.67** | **23.69** | **3.94** | **4.69** | **5.33** | **5.73** |
| CPMA | 9.20 | 12.88 | 14.96 | 17.22 | 1.18 | 1.96 | 2.55 | 3.09 |
| CPMA + Connect. | 8.67 | 12.39 | 14.45 | 16.88 | 1.17 | 1.96 | 2.58 | 3.13 |

**Table 5-3:** Noise sensitivity results on Animal2000. The table shows the average Hausdorff distance and Dubuisson-Jain dissimilarity for different noise levels (5-20) over each element of the dataset. The best and worst methods are highlighted to facilitate the comparison to the CPMA.

compared with the Hausdorff distance. In fact, for some methods, the metric decreases when the noise level increases. We attribute this behavior to the outlier sensibility of the Hausdorff distance. We complete the noise stability analysis showing some examples of the MAT computed with our methodology, in comparison with the MAT computed using other methods in the comparative study. Figure **5-5** shows such comparisons for all the datasets above mentioned datasets.

### 5.2.3 Sensitivity to Rotations

In the continuous framework, it is well known that the medial axis is a rotation equivariant shape representation. More precisely, if we denote by $R$ the rotation matrix around the origin; the rotation equivariance property states that $\mathbf{MAT}(R(\Omega)) \approx R(\mathbf{MAT}(\Omega))$. Regardless of the shape $\Omega$ and the rotation $R$.

Nevertheless, we can experimentally measure the dissimilarity between $\mathbf{MAT}(R(\Omega))$ and $R(\mathbf{MAT}(\Omega))$

| Method | Hausdorff | | | | | Dissimilarity | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 2 | 4 | 6 | 8 | 10 | 2 | 4 | 6 | 8 | 10 |
| 3D thinning | **4.20** | **3.61** | **3.25** | **3.03** | **2.90** | 6.30 | 7.80 | 8.58 | 9.07 | 9.41 |
| TEASAR | 7.76 | 7.36 | 6.39 | 6.24 | 5.92 | 1.09 | 1.55 | 1.98 | 2.44 | 2.80 |
| CPMA | 11.66 | 14.10 | 11.83 | 15.52 | 12.46 | 0.91 | 1.02 | 0.97 | 1.44 | 1.27 |
| CPMA + Connect. | 3.99 | 4.54 | 11.25 | 13.95 | 12.06 | **0.40** | **0.43** | **1.43** | **1.81** | **1.51** |

**Table 5-4:** Noise sensitivity results on Groningen benchmark. The table shows the average Hausdorff distance and Dubuisson-Jain dissimilarity for different noise levels (5-20) over each element of the dataset. The best result is highlighted in bold font for comparison.

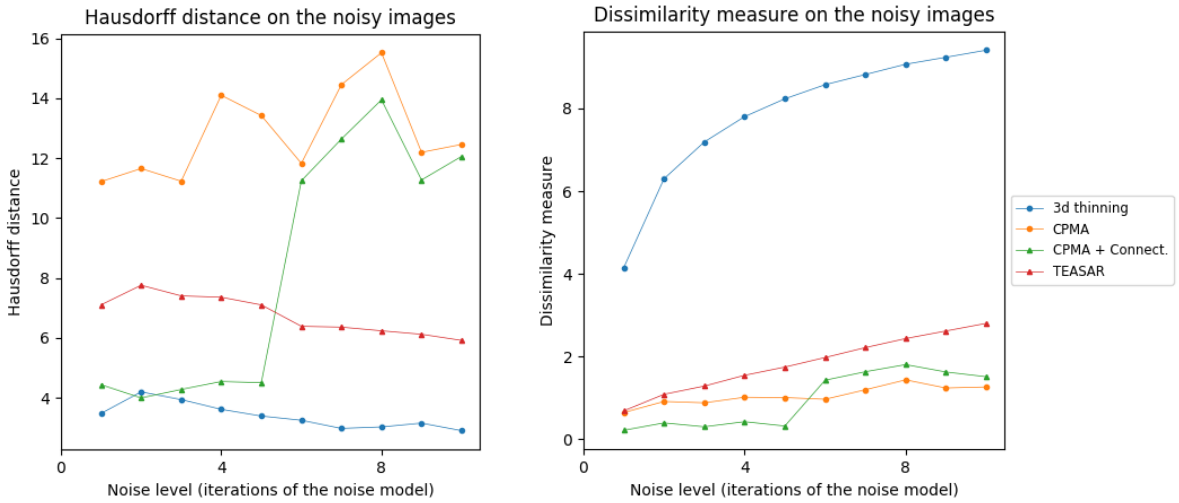



**Figure 5-4:** Noise sensitivity results on Groningen Benchmark dataset. The figure shows the Hausdorff distance (left) and the Dubuisson-Jain dissimilarity (right) for all the methods in table **5-4**.

for different instances, and different definitions of the medial axis transform. The lower this dissimilarity, the more stable the method is under rotation. We conducted experiments on Kimia216, Animal2000, and the Groningen Benchmark by inducing a set of controlled rotations on each element of every dataset. Later, we computed the Hausdorff distance and the Dubuisson-Jain dissimilarity between $\mathbf{MAT}(R(\Omega))$ and $R(\mathbf{MAT}(\Omega))$ for rotation angles varying from 0 to 90 degrees by 3 degrees steps in 2D. In the 3D case, we varied the experiments for computational efficiency. We induced azimuthal rotations (around z-axis) and elevation rotations (around y-axis) up to 90 degrees, but at intervals of 18 degrees.

The rotation sensitivity analysis on the Kimia216 dataset is summarized in table **5-5** and figure **5-6**. The results show that the CPMA and the CPMA with connectivity enforcement curves lie near the average of the rest of the methods achieving state-of-the-art performance and even surpassing some of them. Notice that when using the dissimilarity metric, both CPMA, the GIMA, the SFEMA, and

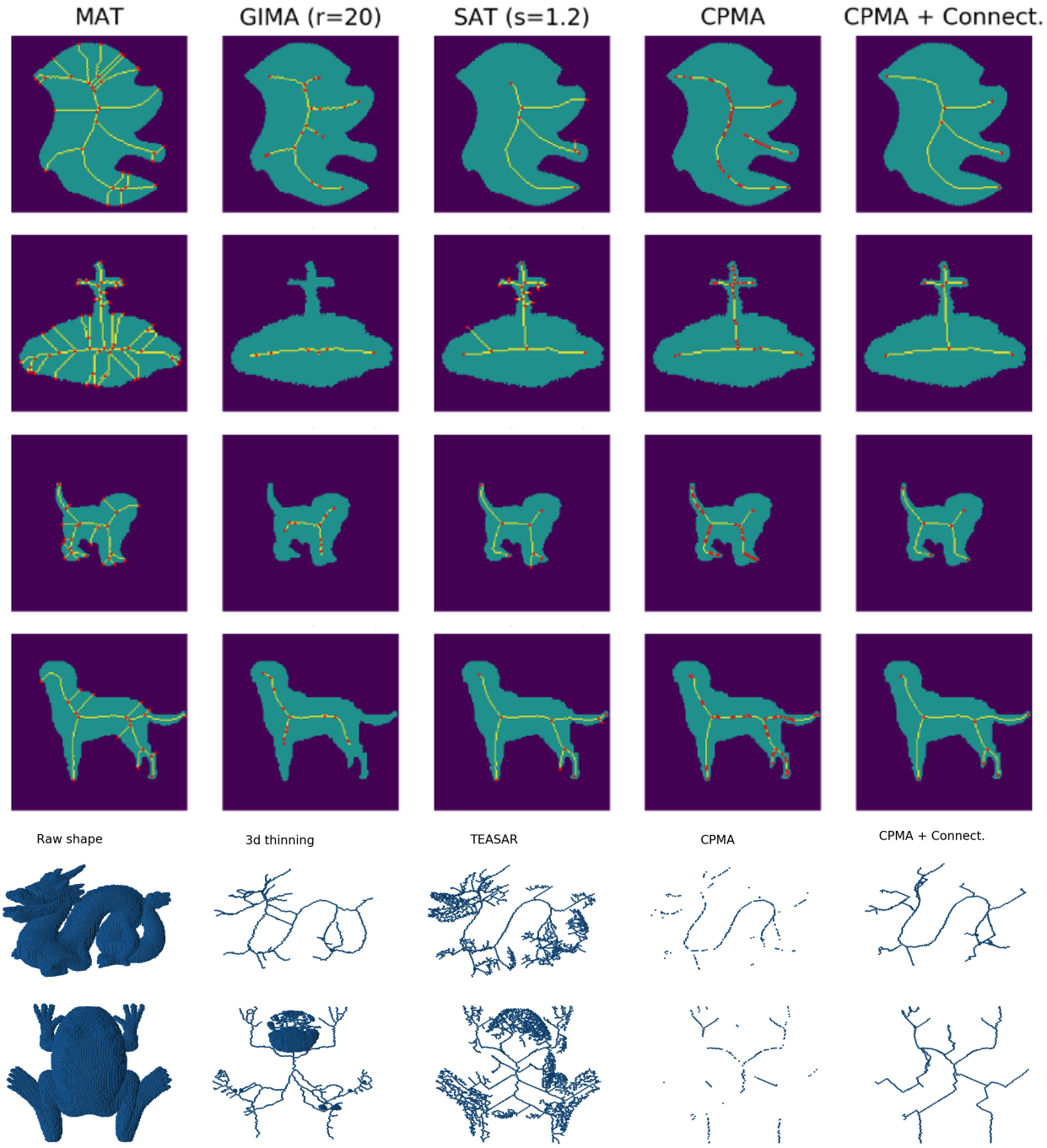



**Figure 5-5:** Skeletonization results. The images show the MAT and the results of four different pruning methods. Rows one and two are objects from Kimia216, rows three and four are from Animal2000, and rows five and six from the Groningen benchmark. Notice how the CPMA and the CPMA + connectivity produces skeletons with less spurious branches while preserving the topology.

the BEMA form a subgroup that performs significantly better compared to the others. Moreover, the performance of these methods oscillates around a value of dissimilarity of around 1 pixel on average. The intuition for this result is that regardless of the rotation, skeletons computed with these methods vary only at one pixel on average. Consequently, we can claim that they exhibit rotation equivariance.

| **Method** | **Hausdorff** | | | **Dissimilarity** | | |
|---|---|---|---|---|---|---|
| | 30º | 60º | 90º | 30º | 60º | 90º |
| MAT | 8.18 | 8.17 | 2.17 | 2.67 | 2.64 | 0.75 |
| Thinning | 7.72 | 7.58 | 8.92 | 2.87 | 2.99 | 1.35 |
| GIMA (r=5) | 6.16 | 6.03 | 5.54 | 1.02 | 1.11 | 0.85 |
| GIMA (r=10) | 5.54 | 6.25 | 5.04 | 0.83 | 1.02 | 0.72 |
| GIMA (r=20) | 3.62 | 3.93 | 3.12 | 1.51 | 0.78 | 1.30 |
| BEMA (theta=90) | 12.35 | 12.76 | 11.72 | 1.31 | 1.60 | 1.06 |
| BEMA (theta=120) | 6.24 | 8.44 | 9.57 | 0.81 | 1.00 | 1.00 |
| BEMA (theta=150) | 9.14 | 10.09 | 10.27 | 1.11 | 1.35 | 1.36 |
| SAT (s=1.1) | 10.84 | 11.93 | 3.96 | 3.22 | 3.34 | 0.97 |
| SAT (s=1.2) | 11.44 | 12.40 | 4.45 | 2.64 | 2.87 | 0.97 |
| SFEMA (s=1.1) | 3.86 | 3.98 | 2.52 | 0.92 | 1.01 | 0.83 |
| **SFEMA (s=1.2)** | **3.68** | **3.66** | **2.08** | **0.82** | **0.88** | **0.80** |
| Poisson skel. (w=0.05) | 12.83 | 13.32 | 8.93 | 3.21 | 3.28 | 1.30 |
| Poisson skel. (w=0.10) | 16.36 | 17.03 | 10.17 | 4.24 | 4.30 | 1.78 |
| **Poisson skel. (w=0.20)** | **18.94** | **19.90** | **9.65** | **5.61** | **5.66** | **2.69** |
| CPMA | 8.63 | 8.65 | 2.42 | 1.18 | 1.33 | 0.70 |
| CPMA + Connect. | 8.51 | 8.36 | 2.72 | 1.22 | 1.39 | 0.75 |

**Table 5-5:** Rotation equivariance results on Kimia216. The table shows the average Hausdorff distance and Dubuisson-Jain dissimilarity for different rotations of each element in the dataset. The best and worse method are highlighted to facilitate the comparison with the CPMA.

We applied the same analysis to the Animal2000 dataset achieving similar results. In this case, the CPMA and the CPMA+connectivity ranked third and fourth, respectively, among all methods when we used the dissimilarity metric. The results for all methods and parameters are presented in table **5-6**. As before, we also present a summary with the best parametrization for each method in figure **5-7** to facilitate the interpretation. Notice that due to the larger number of objects in the Animal2000 dataset, the curves for every method appear to be smoother, highlighting stability across different rotation angles and shapes.

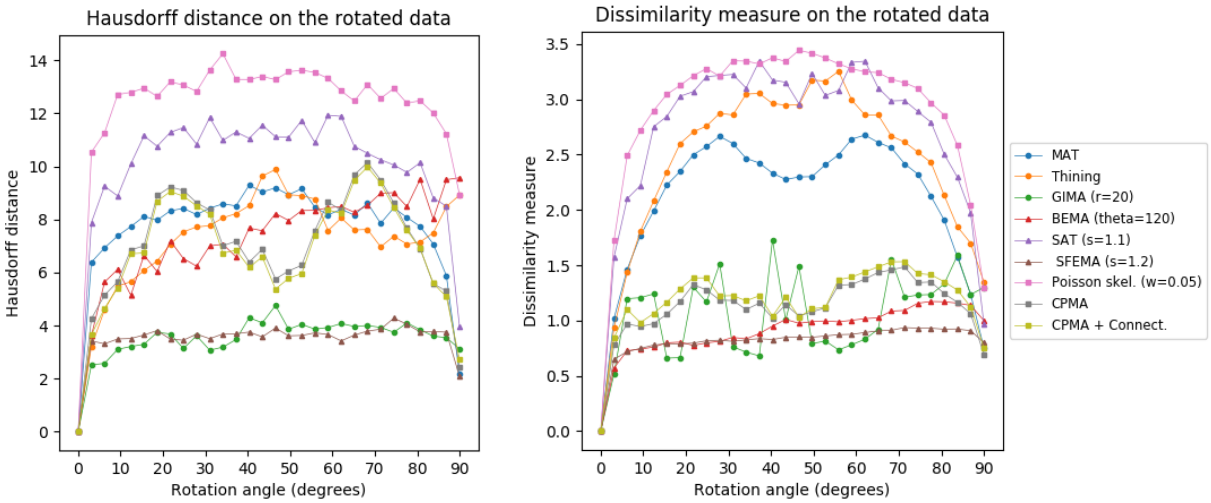



**Figure 5-6:** Rotation equivariance results on Kimia216 dataset. The top row shows the Hausdorff distance and Dubuisson-Jain dissimilarity for all the methods in table **5-5**. The bottom row shows only the best 5 results for clarification.

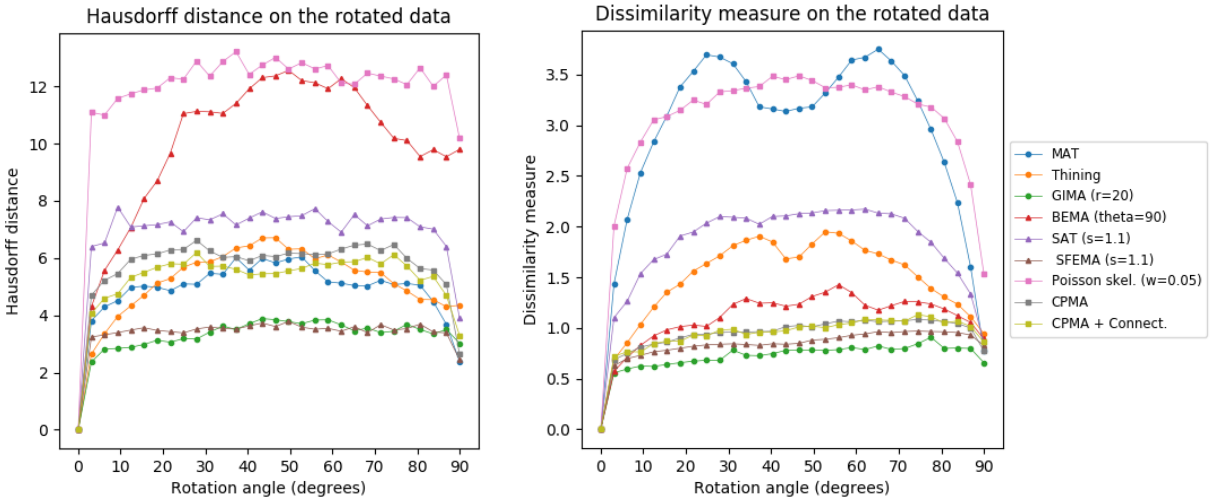



**Figure 5-7:** Rotation equivariance results on Animal2000 dataset. The top row shows the Hausdorff distance and Dubuisson-Jain dissimilarity for all the methods in table **5-5**. The bottom row shows only the best 5 results for clarification.

Finally, we ran the rotation sensitivity analysis on the 3D datasets, and summarize the results in figure **5-8**. The image shows the four 3D skeletonization methods we used in our study for combinations of azimuthal and elevation angles. This figure illustrates how both the Hausdorff distance and the dissimilarity get higher when the rotation becomes more extreme, except in the case of CPMA + connectivity. We believe this behavior is due to the connectivity enforcement mitigating the gaps in the skeletons, and reducing the metrics.

| Method | Hausdorff | | | Dissimilarity | | |
|---|---|---|---|---|---|---|
| | 30° | 60° | 90° | 30° | 60° | 90° |
| MAT | 5.08 | 5.17 | 2.38 | 3.67 | 3.64 | 0.77 |
| Thinning | 5.85 | 6.11 | 4.35 | 1.71 | 1.86 | 0.94 |
| GIMA (r=5) | 5.58 | 5.54 | 5.62 | 0.96 | 1.08 | 0.83 |
| GIMA (r=10) | 5.42 | 5.50 | 4.29 | 0.78 | 0.88 | 0.74 |
| **GIMA (r=20)** | **3.17** | **3.84** | **3.02** | **0.68** | **0.81** | **0.66** |
| BEMA (theta=90) | 11.12 | 11.92 | 9.80 | 1.10 | 1.35 | 0.89 |
| BEMA (theta=120) | 5.45 | 6.10 | 6.80 | 0.77 | 0.90 | 0.86 |
| BEMA (theta=150) | 7.60 | 8.88 | 9.91 | 1.13 | 1.36 | 1.40 |
| SAT (s=1.1) | 7.41 | 7.27 | 3.90 | 2.10 | 2.16 | 0.86 |
| SAT (s=1.2) | 9.82 | 9.62 | 4.85 | 1.77 | 1.90 | 0.95 |
| SFEMA (s=1.1) | 3.52 | 3.54 | 2.45 | 0.84 | 0.93 | 0.83 |
| SFEMA (s=1.2) | 3.54 | 3.63 | 2.26 | 0.77 | 0.87 | 0.82 |
| Poisson skel. (w=0.05) | 12.88 | 12.72 | 10.18 | 3.33 | 3.40 | 1.54 |
| Poisson skel. (w=0.10) | 15.02 | 15.41 | 10.58 | 3.67 | 3.89 | 1.89 |
| **Poisson skel. (w=0.20)** | **16.83** | **17.20** | **8.67** | **4.07** | **4.33** | **1.85** |
| CPMA | 6.62 | 6.14 | 2.64 | 0.96 | 1.06 | 0.77 |
| CPMA + Connect. | 6.19 | 5.77 | 3.30 | 0.98 | 1.05 | 0.86 |

**Table 5-6:** Rotation equivariance results on Animal2000. The table shows the average Hausdorff distance and Dubuisson-Jain dissimilarity for different rotations of each element in the dataset. The best and worse method are highlighted to facilitate the comparison with the CPMA.

.

### 5.2.4 Hyper-parameter Selection

Many medial axis pruning methods depend on hyper-parameters to work correctly. These parameters usually have a physical meaning in the context of the object whose skeleton we sought to estimate. Often, the parameters are distances or angles formed between points inside the object. Some other works also create score function like ours intending to use its values as a filter parameter to remove individual points form the MAT, hoping to reduce the number of spurious branches. However, in most cases, such parameters are subject to factors like resolution or scale. Hence, we conducted another experiment to test the sensitivity of the CPMA to the pruning parameter $\tau$.
We ran additional experiments for different scale factors of the object. Figure **5-9** shows the average of a reconstruction metric vs the selected value of $\tau$. We compared an object $\Omega$ against

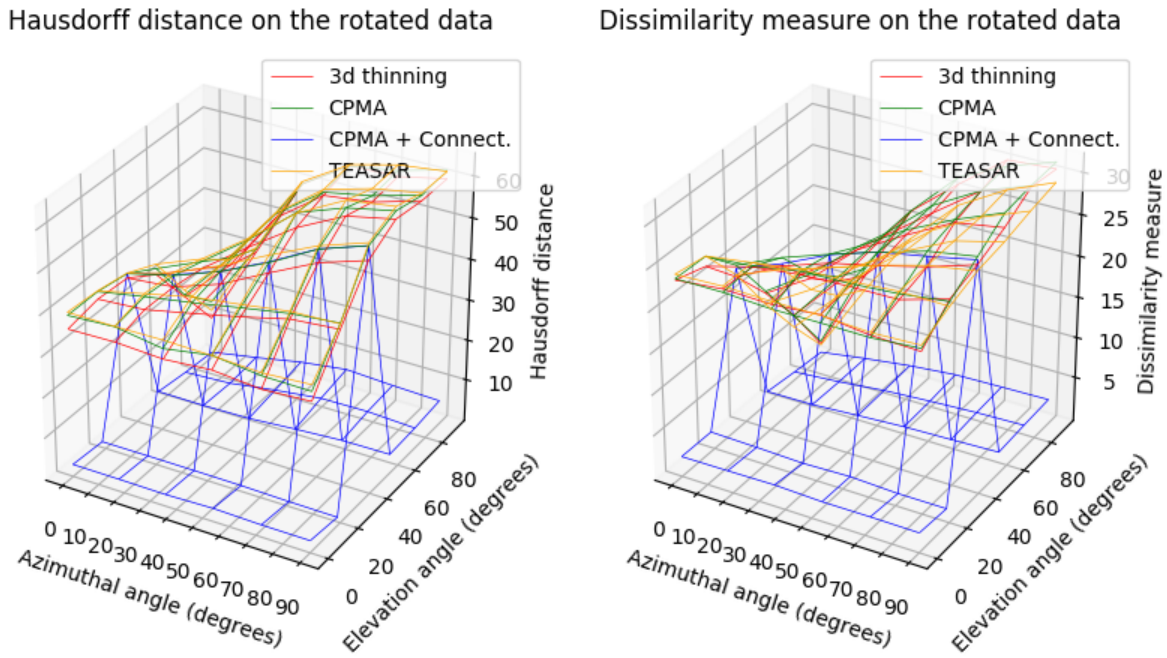



**Figure 5-8:** Rotation equivariance results on Groningen Benchmark dataset. The top row shows the Hausdorff distance and Dubuisson-Jain dissimilarity for all the methods in table **5-5**. The bottom row shows only the best 5 results for clarification.

its reconstruction $\hat{\Omega}$ using the Jaccard Index overall images in Kimia216 dataset. High values of $\tau$ deteriorate the reconstruction, while lower values do not prune enough spurious branches. From the figure, we can infer that values around $\tau = 0.5$ offer a good trade-off between reconstruction and branch pruning. Moreover, around this value, the standard deviation reaches its minimum value suggesting optimal performance regardless of the object. Because the value of $\tau$ is stable for different scale factors, we conclude that scale does not affect the selection of the threshold.
In the next chapter, we will present experiments in which we use the CPMA + connectivity as input shape representation for our classification and retrieval pipeline. We will derive some features from our skeletonization, and then train a machine learning approach to predict the class of each element.

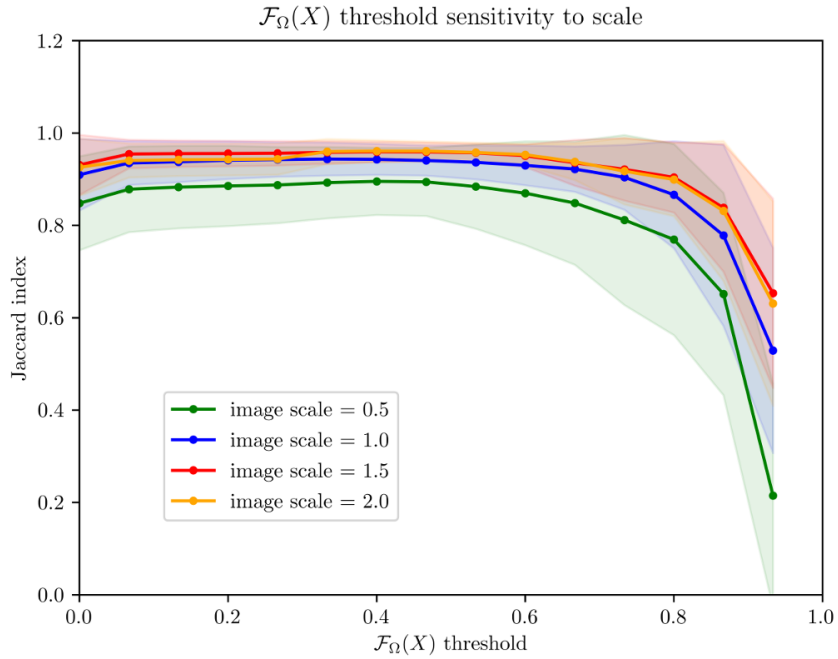



**Figure 5-9:** Sensitivity Analysis of threshold $\tau$ to different scales of the input images. The graph shows the average Jaccard index of the reconstructed shape w.r.t the original object for CPMAs computed with different values of $\tau$. Higher values of the threshold lead to less spurious branches. We also show the standard deviation error bands.

# 6 Shape-based Object Classification and Retrieval

After developing our method for robust skeletonization, the next step is to use skeletons produced with it into a machine learning pipeline to classify objects into previously defined categories. Hence, in this chapter, we describe in detail the set of features that we engineered and the machine learning architecture that was designed to classify each element. We based our features on the Chordiogram (Toshev, 2011), which is a rotation and translation invariant shape descriptor capturing global geometric relationships between elements of the object boundary. We also offer a justification of why our features achieve isometric invariance, which is one of our main goals. A set of classification and retrieval experiments on MPEG7 and ModelNet-40 dataset was conducted to test our approach in both 2D and 3D, comparing the results with state-of-the-art methods.

## 6.1 Machine Learning Pipeline

### 6.1.1 Skeleton Based Feature Extraction

To extract a set of features from the skeletal representation of $\Omega$ we used the **Chordiogram** defined by Toshev (2011). The chordiogram is a rotation and translation invariant shape descriptor that captures global geometric relationships between points of the object boundary.

To define the chordiogram, consider a pair of boundary edges $p, q \in \delta\Omega$. We will call such the pair $(p, q)$ a chord. One can define various features describing the geometry of the chord, which we will denote by $f_{pq} \in \mathbb{R}^d$. The chordiogram is then a k-dimensional histogram of $f_{pq}$ over all the chords. The chordiogram can capture the invariant properties of $\Omega$ by carefully choosing $f_{pq}$. This happens because isometric transformations apply equally to all points in the object. e.g., if we set $f^0_{pq}$ as the euclidean distance between $p$ and $q$, and set $R$ as a rotation around the origin of coordinates.

We derived a set of features inspired by the chordiogram and adapted them to work with the information provided from the skeleton of the object. The skeletons, as we defined in earlier chapters, are shape representations that comprise the object into a set of points and radius, $(X, r)$. Moreover, the skeleton of the object can also be seen as graph $\mathcal{G}$, where the nodes are points in which three or more branches converge. We used this definition to design three new skeletal features by creating chords between the nodes of $\mathcal{G}$.

Let us define $n_i$ and $n_j$ as two distinct nodes of $\mathcal{G}$. We called the first skeletal feature $\gamma_{i,j}$, and

define it as the ratio of the euclidean distance and the geodesic distance between $n_i$ and $n_j$.

$$\gamma_{i,j} = \frac{d_{euc}(n_i, n_j)}{d_{geo}(n_i, n_j)}, \tag{6-1}$$

where, $d_{euc}$ is the Euclidean distance between the $\mathbb{R}^{\kappa}$ coordinates of both nodes, and $d_{geo}$ is the geodesic distance through $\mathcal{G}$ between $n_i$ and $n_j$.
We call the two remaining skeletal features $\pi_i$ and $\pi_j$, and define them as the Euclidean distance from $n_i$ and $n_i$ to the closest boundary point, respectively. This formulation of skeletal features allows the characterization of an object by their relationships among pairs of elements of its topological structure. The MAT of the shape represents such a topological structure.
Table **6-1** show a list of the features that are part of the original formulation of the chordiogram, and the skeletal features formulated in the present study. This invariance properties of the skeletal features will be discussed in the next subsection.

| | **Feature Name** | **Invariance** | | | **Formulation** |
|---|---|---|---|---|---|
| | | **Rot.** | **Scale** | **Trans.** | |
| $l_{pq}$ | Chord length | yes | no | yes | Regular chord |
| $l_p$ | Distance to the center | yes | no | no | Regular chord |
| $\psi_{pq}$ | Chord orientation | no | yes | yes | Regular chord |
| $\theta - \psi_{pq}$ | Relative normal | yes | yes | yes | Regular chord |
| $\gamma_{i,j}$ | Ratio between $d_{euc}$ and $d_{geo}$ of nodes $n_i$ and $n_j$ | yes | yes | yes | Skel. chord |
| $\pi_i$ | Distance from $n_i$ to $\delta\Omega$ | yes | no | yes | Skel. chord |
| $\pi_j$ | Distance from $n_j$ to $\delta\Omega$ | yes | no | yes | Skel. chord |

**Table 6-1:** Chordiogram features and their invariance. The table shows all the features derived from the chordiogram with their respective invariance.

### 6.1.2 Invariant Properties of the Skeleton-Based Features

All of our skeletal features are invariant to scale, translation, and rotation. Therefore, they are suitable to be used in our machine learning pipeline to classify objects by their shape.

**Translation Invariance** If we define point $p_i, p_j \in \mathbb{R}^m$ as the coordinates of $n_i$ and $n_j$, and $l$ as an arbitrary vector also in $\mathbb{R}^m$, we can show that

$$d_{euc}(p_i + l, p_j + l) = ||p_i + l - p_j - l|| = ||p_i - p_j||.$$

The same result holds for $d_{geo}$; hence, proving the translation invariance of $\gamma_{i,j}$. We can reason similarly with $\pi_i$ and $\pi_j$ because of all the points in $\Omega$ translate uniformly by $l$.

**Rotation Invariance** Recalling that a transformation with a rotation matrix $R$ is a unitary transformation, we can show that our features have rotation invariance. As it is the case with the translation invariance, the distance between two points is not affected by rotations since

$$\begin{aligned} d_{euc}(Rp_i, Rp_j) =& ||Rp_i - Rp_j|| \\ =& ||R(p_i - p_j)|| \\ =& \sqrt{(p_i - p_j)^T R^T R (p_i - p_j)} \\ =& \sqrt{(p_i - p_j)^T (p_i - p_j)} \\ =& d_{euc}(p_i, p_j). \end{aligned}$$

**Scale Invariance** The feature $\gamma_{i,j}$ is approximately scale-invariant because it is the ratio of two distances that grow equally with a scale factor $s$. However, $\pi_i$ and $\pi_j$ are non-bounded distances that depend on the dimensions of the object $\Omega$. To enforce the scale invariance, we normalize the distances by the maximum distance value over all nodes of $\mathcal{G}$,

$$\hat{\pi}_i = \frac{\pi_i}{\pi_{max}}.$$

Using the above derivations, we can extend the invariance of the skeletal features to all isometric transformations.

### 6.1.3 Deep Learning Architecture

Our machine learning pipeline uses a deep learning architecture based on PointNet++ (Qi et al., 2017b). PointNet++ is a CNN model designed to learn features on orderless point clouds. It is permutation invariant, which means that the order of the points of the input features does not affect the task the model is performing. PointNet++ is an extension of an earlier model from the same authors, PointNet (Qi et al., 2017a). The differentiating factor lies in the fact that the latter version enforces spatial localities of the input point set, as a means to preserve well-defined distance metrics on the object such as Euclidean distance or geodesic distances.
PointNet++ builds a hierarchical grouping of points and progressively abstract larger local regions along with the hierarchy, layer upon layer. After stacking several of these layers, the architecture ends up with a smaller set of points representing the underlying structures of the object, each point equipped with $K$-dimensional features vectors.
This hierarchical structure is composed of a number of set abstraction levels that include: a sampling layer, a grouping layer, and a pointNet layer. The Sampling layer selects a set of points from input points, which defines the centroids of local regions. The grouping layer constructs local region

sets by finding "neighboring" points around the centroids. PointNet layer uses a mini-PointNet to encode local region patterns into feature vectors. If the final goal is classification, the features pass through an MLP to be trained with a softmax as the classification loss function. The entire architecture can be seen in Figure **6-1**.

In this study, we branched PointNet++ in two such that we could process the contour and the skeletal data separately as it is depicted in figure **1-1**. We call the first one the point cloud branch since it takes a point cloud sampled from either a 3D surfaces or a 2D contour as input. The second branch is called the skeletal branch, and it takes a list of chords extracted from the joints of the skeleton as its input.

PointNet++ is generalizable to the chord space because a list of chords can be seen as a point cloud in the m-dimensional space defined by the number of features of the chords. Despite this, the contour data has a geometric meaning, e.g., a sampling of the boundary of the object, while the skeletal data is just a set of $n$ points features embedded in $\mathbb{R}^m$. The CNN needs to learn different classes of features from each input type, which is why we must process them separately.

The result of each branch is a one-dimensional feature vector of size 512. We concatenate both features vectors into a new one that we call $\phi$. This new feature vector contains information from both the surface point cloud and the skeletonization. Finally, we pass $\phi$ through a fully connected MLP and train the whole model for classification in an end-to-end fashion.

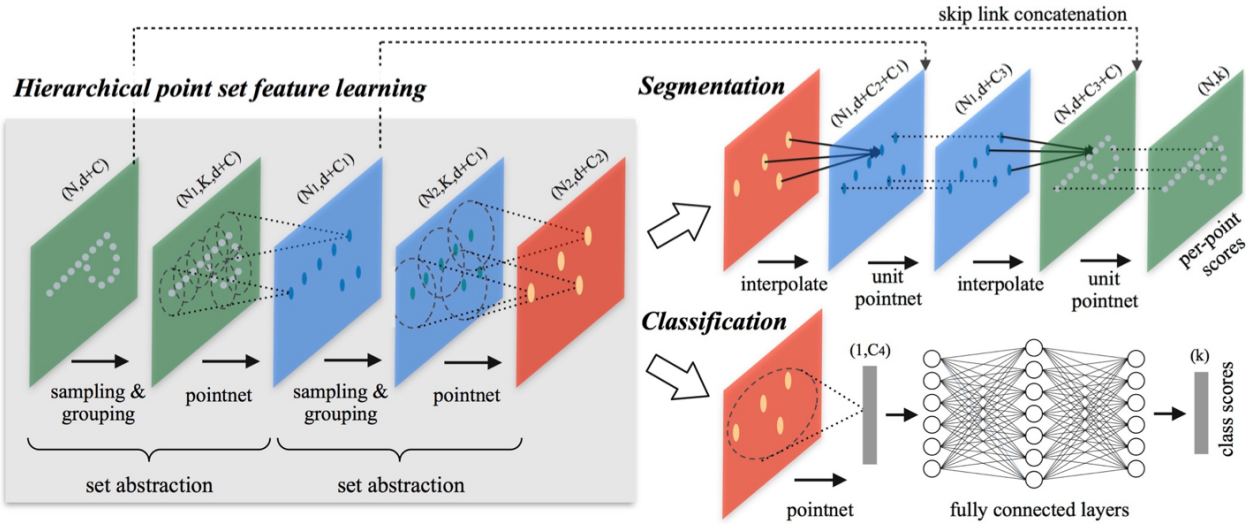



**Figure 6-1:** PointNet++ Architecture.

After the network is trained, the features in $\phi$ that are learned with the machine learning architecture are used as feature vectors for shape retrieval.

## 6.2 Experiments and Results

We conducted a set of shape classification and shape retrieval experiments on the MPEG7 and ModelNet40 datasets to test our methodology. We used the per-instance accuracy to test the

classification performance, and the Bull's eye score to measure the retrieval performance. In this section, we offer implementation details of the entire pipeline and discuss the results.

### 6.2.1 Training

Our pipeline is generalizable from 2D to 3D, meaning that the deep learning model does not change in its mathematical design. However, when implementing the code, some considerations, such as the format of the dataset, needed to be considered.
For every case in 2D and 3D, we used two models: 1) a baseline using plain PointNet++ as specified in the original paper, 2) our methodology including the robust skeletonization designed and explored in chapter 5. Moreover, three modes of training were considered for testing the rotation invariance of our classification pipeline.
In the 2D case, an image can be in either the canonical pose as it is presented in the dataset or can have rotations around the origin ($\mathbf{SO}(2)$ rotations). Therefore, the three modes of training for the 2D case are: 1) training and testing in canonical pose (c/c), 2) training and testing with arbitrary rotations (SO(2)/SO(2)), and (3) training in canonical pose and testing with arbitrary rotations (c/SO(2)). In 3D, the situation is similar. However, because of the complexity of the stated problem in 3D, we used azimuthal rotations instead of the canonical pose. The three training modes in 3D are then: (1) training and testing with azimuthal rotations (z/z), (2) training and testing with arbitrary rotations (SO(3)/SO(3)), and (3) training with azimuthal and tested with arbitrary rotations (z/SO(3)).
We trained each model using the ADAM optimizer during 350 epochs with an initial learning rate of $10^{-3}$. We used data augmentation for training by adding random jitter, point permutations, and small shifting to the input features. Note that although our learned representation should be rotation invariant, augmenting the inputs with rotations is still beneficial thanks to interpolation and sampling effects.
The results of all our experiments are summarized in tables **6-2** and **6-3**. They will be discussed in the following subsections, along with more details about each type of experiment.

| **Method** | **Mode** | **Accuracy** | **Bull's eye score** |
|---|---|---|---|
| PointNet++ baseline | c/c | 88.92 % | 0.8369 |
| | c/$\mathbf{SO}(2)$ | 34.29 % | 0.8407 |
| | $\mathbf{SO}(2)$/$\mathbf{SO}(2)$ | 78.93 % | 0.8018 |
| Our method | c/c | **95.00 %** | **0.8418** |
| | c/$\mathbf{SO}(2)$ | 36.07 % | 0.8315 |
| | $\mathbf{SO}(2)$/$\mathbf{SO}(2)$ | 80.00 % | 0.7457 |

**Table 6-2:** Classification and retrieval results of our methodology in 2D. We used PointNet++ as a baseline to compare the results of our methodology.

| Method | Mode | Accuracy | Bull's eye score |
|---|---|---|---|
| PointNet++ baseline | z/z | **89.30** % | 0.6228 |
| | z/**SO**(3) | 47.37 % | 0.6199 |
| | **SO**(3)/**SO**(3) | 71.63 % | 0.5549 |
| Our method | z/z | 84.44 % | **0.7910** |
| | z/**SO**(3) | 54.21 % | 0.7872 |
| | **SO**(3)/**SO**(3) | 53.61 % | 0.7751 |

**Table 6-3:** Classification and retrieval results of our methodology in 3D. We used PointNet++ as a baseline to compare the results of our methodology.

### 6.2.2 Classification

The classification was done by adding a softmax layer at the end of our deep learning model. For each input object that passes through the model, it returns a vector $p$ of dimension $K$, where $K$ is the number of classes. Each element of this vector $p_i$ is the probability of the object belonging to the class $i$. As is common in classification, we trained our model by minimizing the softmax cross-entropy between $p_i$ and the ground truth vector $\hat{p}$, which is sparse with only one value different from zero, $\hat{p}_i = 1$, when the object belongs to the class $i$. To evaluate the classification performance, we computed the accuracy per instance by taking the ratio of all of the elements correctly classified over the total number of elements.

The classification results are summarized in tables **6-2** for 2D classification on the MPEG7 dataset; and in table **6-3** for the 3D ModelNet40 dataset. We present results for every one of the training modes described in the previous subsection.

From table **6-2** is possible to see how the classification accuracy is higher when our method is employed. We reached a value of 95 % accuracy, which means that 95 out of 100 objects were well classified into one of the 70 classes of MPEG7. We got this value when we trained the model on the c/c mode. With the modes c/**SO**(2) and **SO**(2)/**SO**(2) we obtained accuracy values of 36.07% and 80% respectively. These values are both approximately 2% higher than their counterpart mode when training on the PointNet++ baseline. We attribute this increase in accuracy to the invariant properties of our skeleton-based features.

The confusion matrices for the models trained with our method are depicted in the first row of figure **6-4**. The figure shows a confusion matrix that is highly populated in its diagonal for the model trained in the mode c/c, following table **6-2**. Similar results occur with mode **SO**(2)/**SO**(2), where we start seeing more disperse values around the confusion matrix that indicate miss-classified elements. However, the main diagonal still contains the majority of the high values in the matrix. Therefore, we conclude that there is no significant confusion between any of the classes. The c/**SO**(2) mode, has high values scattered around, in concordance with table **6-2**.

Unfortunately, in the 3D case, our methodology did not achieve as positive results as in the 2D case. The performances for all methods of training were below the baseline, except in the z/**SO**(3) bases.

However, the z/**SO**(3) mode must not be entirely trusted since the model training did not converge, as can be observed in its loss function progression in figure **6-3**. The figure also shows the same diverging pattern for the training mode **SO**(3)/**SO**(3). The only mode that whose loss function significantly approached 0 was the z/z; however, its results are not competitive when compared with the baseline or with the state-of-the-art methods of 3D classification on the ModelNet40 dataset (see table **6-4**). The confusion matrices for all tests of the 3D experiments are depicted in the second row of figure **6-4**.

| **Method** | **Author** | **Input type** | **Acc.** | **Params.** | **Inp. size** |
|---|---|---|---|---|---|
| VoxNet | Maturana and Scherer (2015) | Voxel-grid | 83.0 | 0.9M | $30^3$ |
| SubVolSup | Qi et al. (2016) | | 88.5 | 17M | $30^3$ |
| SubVolSup MO | Qi et al. (2016) | | 89.5 | 17M | $20\text{x}30^3$ |
| RotationNet 20x | Kanezaki (2016) | 2D Image | **92.4** | 58.9M | $20\text{x}224^2$ |
| MVCNN 12x | Su et al. (2015) | | 89.5 | 99M | $12\text{x}224^2$ |
| MVCNN 80x | Su et al. (2015) | | 90.2 | 99M | $80\text{x}224^2$ |
| PointNet | Qi et al. (2017a) | Point cloud | 89.2 | 3.5M | 3x2048 |
| PointNet++ | Qi et al. (2017b) | | 89.3 | 1.7M | 3x1024 |
| Spherical CNN | Esteves et al. (2018a) | Mesh | 88.9 | 0.5M | $2\text{x}64^2$ |
| **Ours** | | | **84.4**$^*$ | 0.8M | 3x1024 |

**Table 6-4:** Classification results on the ModelNet40 dataset. We compared the accuracy of our results with several state-of-the-art methods in different categories.

### 6.2.3 Retrieval

In order to run the shape retrieval experiments, we needed to describe every object in the datasets with a set of $m$ features arranged in a vector. The idea behind retrieval is that objects that are close in this $\mathbb{R}^m$ space should belong to the same class, while objects of different classes should be far apart among them. To do this, we used the feature vector $\phi$ of 1024 elements, which is the concatenation of the output of both branches of our deep learning model. The retrieval experiments consisted of comparing every object (query object) in the dataset with the rest of the objects and selecting the $k$ most similar ones. A pair of similar objects means that the euclidean distance $d(\phi_i, \phi_j)$ is small. Among the $k$ most similar objects, those who belong to the same class as the query object are used to compute a retrieval metric. In our case, we used the Bull's eye score because of its simplicity.

The last column in tables **6-2** and table **6-3** show the Bull's eye score computed on the MPEG7 and ModelNet40 dataset respectively, for each mode of training. The Bull's eye score is computed as a number in the range $[0, 1]$, where 0 means total failure when retrieving similar objects from

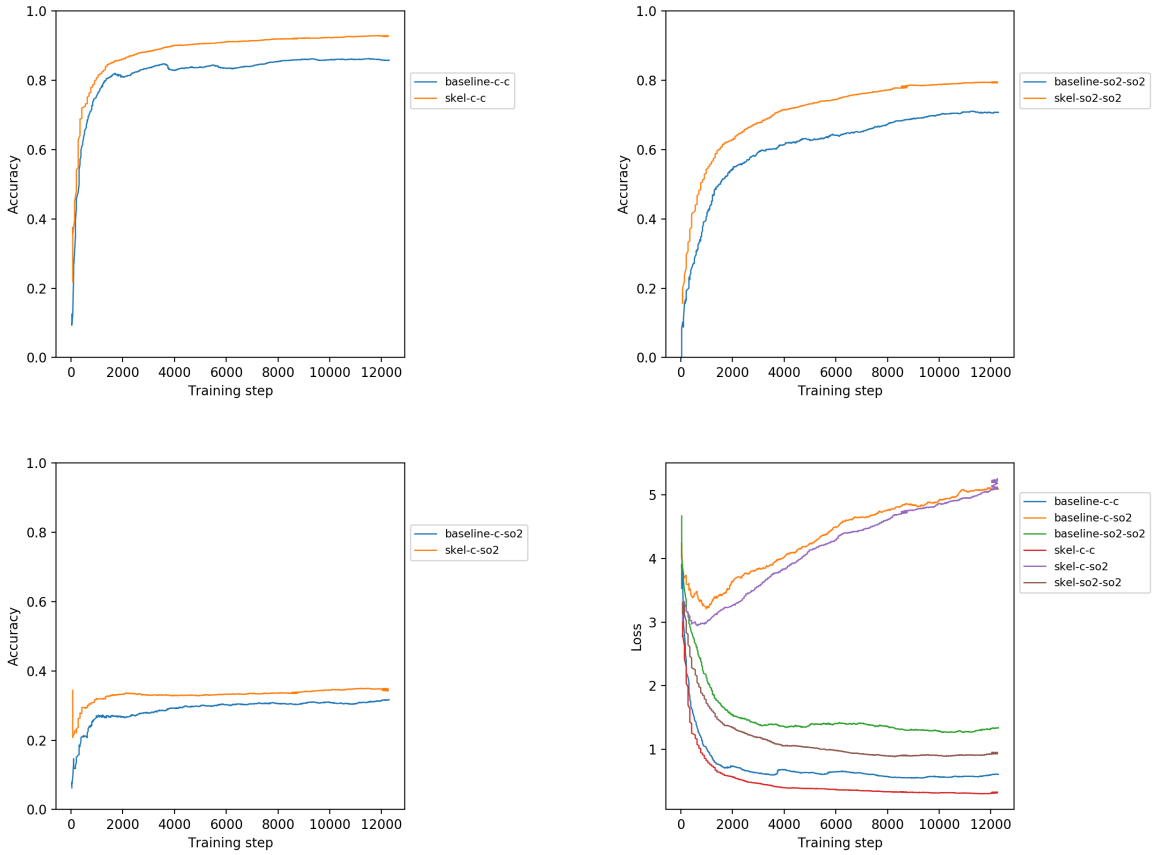



**Figure 6-2:** Classification training progress on the MPEG7 dataset. The images show the training progress step by step and up to 250 epochs. From left to right and top to bottom we preset: test accuracy for the c/c mode, test accuracy for the **SO**(2)/**SO**(2) mode, test accuracy for the c/**SO**(2) mode, and the classification loss for all modes.

the dataset, and 1 means that retrieved most similar objects were always of the same class as the query object.

We conducted two retrieval tests: 1) on the MPEG7 2D dataset and 2) on the ModelNet40 3D dataset. For the 2D experiments, we compared the retrieval results with the state-of-the-art in shape retrieval. These results are presented in table **6-5**. In this table, a set of methods with the best performance in the literature were selected for comparison. The average score overall methods in table **6-5** is 0.8300. The best performing method was the Locally constrained diff. (Yang et al., 2009) that achieved a score of 0.9332. In comparison, our methodology achieved a Bull's eye score of 0.8418, which is slightly above average. Although our method could not surpass the Locally constrained diff., we consider that our results are still competitive and have the potential for improvement.

The Bull's eye score was also computed over the ModelNet40 dataset. Results in table **6-3** show superior retrieval performances of more than 15%. In all tests conducted with our model, the Bull's eye score achieved better results than their counterparts in the baseline. We see this as evidence

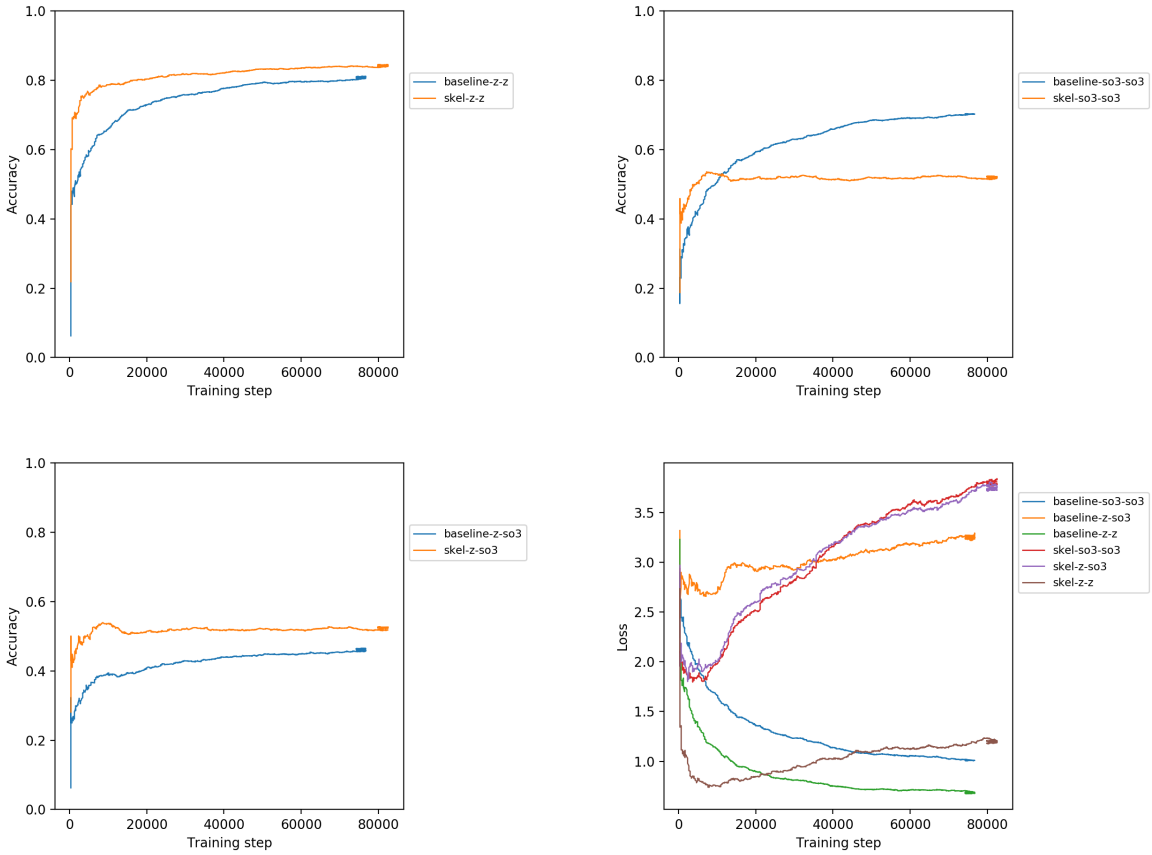
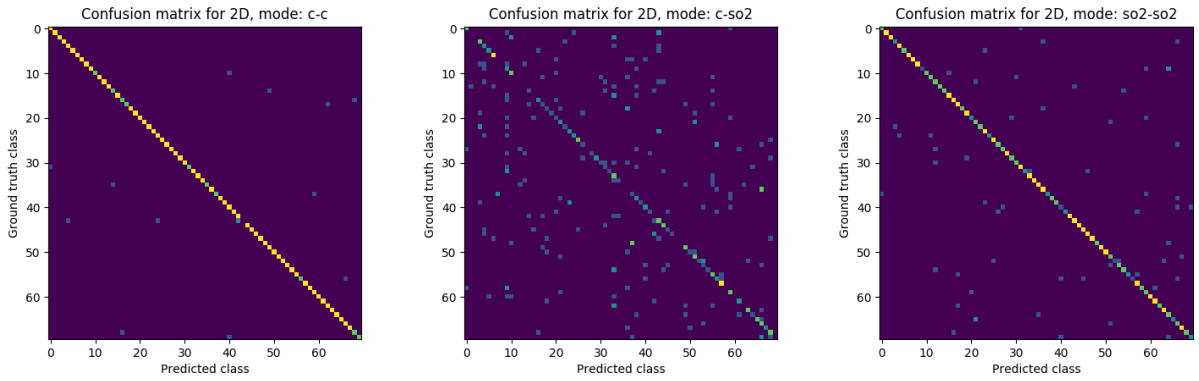



**Figure 6-3:** Classification training progress on the ModelNet40 dataset. The images show the training progress step by step and up to 250 epochs. From left to right and top to bottom we preset: test accuracy for the z/z mode, test accuracy for the **SO**(3)/**SO**(3) mode, test accuracy for the z/**SO**(3) mode, and the classification loss for all modes.

that the skeletal features are improving the shape representation.

Additionally, we want to highlight two key conclusions that arise from our results. First, notice how the retrieval scores for our method are roughly the same regardless of the rotation mode of the train and test partition of the dataset, even when the classification accuracy differs significantly. Such scores suggest isometric invariance induced by our skeletal features. The second key result is that our methodology managed to achieve performances of around 0.78 in a more complex problem, namely 3D shape classification, as opposed to 2D shape classification.

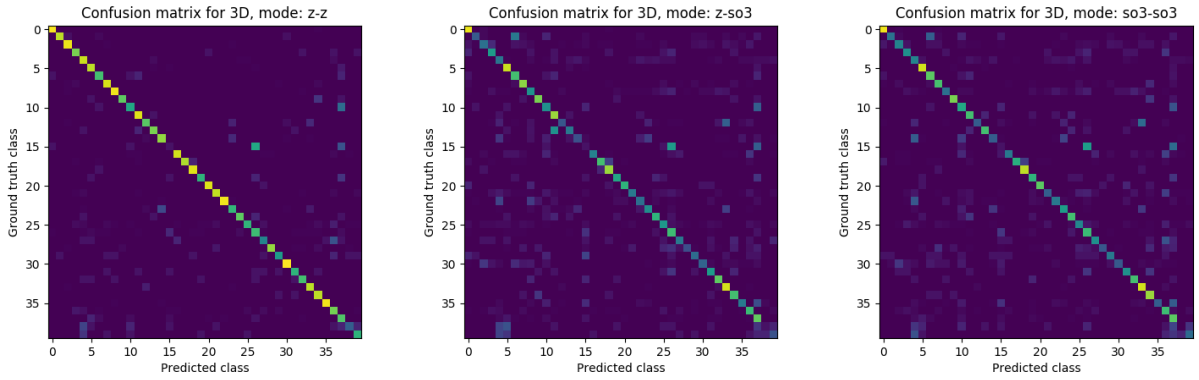
In the next chapter, we will conclude this study offering some final remarks, including conclusions of the skeletonization and the classification methods. Additionally, we will introduce potential future work to continue the line of research in this dissertation.




**(a)** Confusion matrix on MPEG7, mode **c**/**c**

**(b)** Confusion matrix on MPEG7, mode **c**/**SO**(2)

**(c)** Confusion matrix on MPEG7, mode **SO**(2)/**SO**(2)




**(d)** Confusion matrix on ModelNet40, mode **z**/**z**

**(e)** Confusion matrix on ModelNet40, mode **z**/**SO**(3)

**(f)** Confusion matrix on ModelNet40, mode **SO**(3)/**SO**(3)

**Figure 6-4:** Confusion matrices of the classification using our methodology for every mode of training. The first row shows the confusion matrices for 2D classification on the MPEG7 dataset. The second row shows the confusion matrices of 3D Classification on ModelNet40.

| **Method** | **Authors** | **Bull's eye score** |
|---|---|---|
| CSS | Mokhtarian et al. (1998) | 0.7544 |
| Vis. Parts | Latecki and Lakamper (2000) | 0.7645 |
| Shape Context | Belongie et al. (2002) | 0.7651 |
| Aligning Curves | Sebastian et al. (2003) | 0.7816 |
| Distance Set | Grigorescu and Petkov (2003) | 0.7838 |
| Optimized CSS. | Mokhtarian and Bober (2003) | 0.8112 |
| Gen. Model | Tu and Yuille (2004) | 0.8003 |
| Chance Prob. | Super (2004) | 0.7936 |
| Multiscale Representation | Adamek and O'Connor (2004) | 0.8493 |
| Prob. Approach | McNeill and Vijayakumar (2005) | 0.7919 |
| Contour Seg. | Attalla and Siy (2005) | 0.8433 |
| Fixed Cor. | Super (2006) | 0.8540 |
| Hierchical Procrustes | McNeill and Vijayakumar (2006) | 0.8635 |
| Shape Tree | Felzenszwalb and Schwartz (2007) | 0.8770 |
| Inner Distance | Ling et al. (2007) | 0.8540 |
| Triangle Area | Alajlan et al. (2008) | 0.8723 |
| Skeletal Context | Xie et al. (2008) | 0.7992 |
| Shape L'AneRouge | Peter et al. (2008) | 0.8525 |
| **Locally constrained diff.** | Yang et al. (2009) | **0.9332** |
| Context-sensitive shape Sim. | Bai et al. (2010) | 0.9161 |
| Symbolic Representation | Shen et al. (2014b) | 0.8592 |
| **Our method** | **-** | **0.8418** |

**Table 6-5:** Shape retrieval results of the experiments on the MPEG7 dataset. State-of-the-art methods are shown ordered by publication year, along with authors for reference. Our method and the best performance in the table are highlighted in bold letters to facilitate the comparison. The average Bull's eye score for the methods in the table is 0.8300.

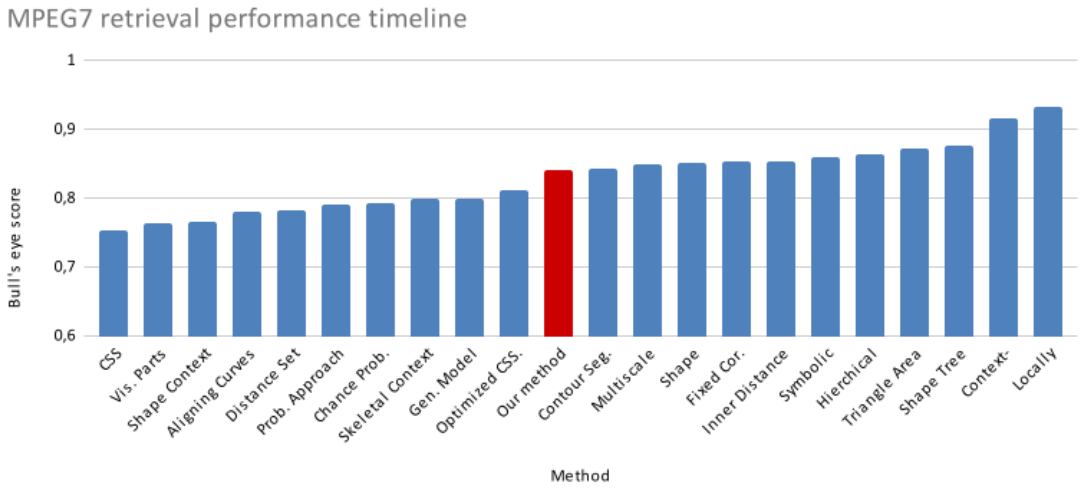



**Figure 6-5:** Shape retrieval timeline on the MPEG7 dataset. The image shows how our method (red) lies within the average of the other state of the art methods.

# 7 Final Remarks

In this dissertation, we studied shape analysis for classification and retrieval. We focused our work on designing a new shape description strategy with invariant properties to isometric transformation; however, we reviewed a large number of different works in the state-of-the-art related to shape representation and description.
We chose the Medial Axis Transform as our shape representation because of its properties that make it invariant to isometries. However, due to its extreme sensitivity to noise, we first formulated a robust skeletonization algorithm capable of estimating the "true skeleton" of an object with fewer spurious branches. We designed a machine learning approach to extract shape features from the medial axis transform, that can be applied to 2D and 3D shapes. We conducted shape classification and retrieval experiments in order to assess the advantages of our approach against state-of-the-art methods.

## 7.1 Conclusions

Our pruning approach shows competitive results compared to the state-of-the-art on pruning methods of the Medial Axis of an object. Results in chapter 5 show that our method achieves significant noise insensitivity, being able to produce stable skeleton even in scenarios with significant perturbations of the contour. Additionally, experiments in chapter 5 produced stable results in the presence of isometric transformations of the object. Through the formulation of the CPMA, we concluded that such equivariance is generalizable to any isometric transformation.
The CPMA developed in this work has other interesting properties. First, it can be efficiently computed in parallel because it depends on an aggregation of reconstructions $\hat{\Omega}$ of the original shape. Each reconstruction is independent of the others, which allows the parallelism. Moreover, we enforce the CPMA to maintain the topology of the original object. The aforementioned occurs because the CPMA is ultimately computed out of the MAT of the reconstructions.
Our results also suggest that our CPMA noise invariant properties generalize across different datasets. This occurs because of the complexity of Animal2000 in terms of the number of shapes and the diversity of them.
In this study, we developed a new skeletonization method and a shape classification architecture. In both cases, our methodology can be easily applied to 2D or 3D, meaning that our proposed methods generalize across dimensions. This is a useful result for the implementation of any application based on shape analysis.
As stated in chapter 3, spectral methods have been gaining attention in the scientific community. Because our work depends on the cosine transform, we can argue that our approach lies in this category, enforcing spectral shape analysis through our results.

We used the equivariance properties of the medial axis transform and the chordiogram definition to design and build a new machine learning pipeline. In the process, we formulated a set of new shape features based on topological skeletons that are invariant to isometric transformations.
Although our classification results did not surpass the state-of-the-art by a large margin, we still consider that we achieved comparable results with our methodology. We theorize that by including a more extensive set of invariant skeletal features, and by using other deep learning architectures, our results can be significantly improved.

## 7.2 Future Work

All of the 3D objects we used in our work were stored as 3D triangular meshes. To compute the CPMA and use the skeletons as input for the machine learning, we voxelized these meshes. Voxelization has two types of issues. First, the resolution starts playing an essential role because low lattices have less representative power to capture small details in the objects; therefore, affecting the overall performance. The second issue is the rotation invariance. When the 3D information is voxelized, the three canonical axes need to be defined beforehand, such as the sides of individual voxels align with them. This decision affects the isometric invariance because, for instance, rotated voxels will not align perfectly with non-rotated voxels. As a result, implementing skeletonization methods that can act directly over the 3D meshes or point clouds sampled from the object's surface is a potential line of future work.
Although we developed an effective algorithm for preserving the topology of the medial axis through CPMA, our methodology is not an efficient implementation. Our algorithm for connectivity enforcement relies on iterative computations of the Dijkstra's algorithm for finding the geodesic path between nodes of two subsets of the skeletal graph. This algorithm does not scale well as the image increases in size because the geodesic distance depends on the size of the image seen as a lattice. Consequently, this methodology could be improved by computing the paths in parallel or by using lookup tables of the already computed paths to reduce execution time.
We consider that our methodology has more potential to deal with non-rigid deformation because of the graph-nature of the medial axis. New features that take into account the skeleton joints and the angles between them can be useful to model deformations like articulation. We believe that engineering new features with these characteristics, or formulating new machine learning approaches able to learn these features, can be beneficial for non-rigid object classification and retrieval.
Feature engineering is fundamental to the application of machine learning and is both difficult and expensive. Although automated feature learning can help to ease such problems, a good initial representation of the data is still crucial. In this study, we presented a set of skeleton-based input features for shape classification that are invariant isometric transformations; however, many other useful features can be extracted from the definition of the MAT and should be considered for future research.
PointNet and PointNet++ are arguably the most widely used CNN model for point clouds. Its mathematical foundations are well supported, and it has been used successfully in different datasets. However, other models also claim to perform well on orderless point clouds. It is an interesting line of future work to test models such as (Huang and You, 2016), or (Wang et al., 2019b) on our set

of skeleton-based features, and compare them with the ones presented in this dissertation.
We tested our skeleton-based features through a classification pipeline and retrieval experiments. Retrieval was conducted computing the Bull's eye score, which is the result of comparing an object with all other objects in a dataset. The value of the Bull's eye score is obtained by counting the number of elements of the same class out of the first $2n_i$ most similar objects. The value of $n_i$ is the number of elements in class $i$ in the dataset. Although this metric reflects whether the retrieval is effective or not, other metrics are also employed in the literature. A line of future work is to explore how well our methodology performs when compared with other retrieval metrics such as precision, recall, mean average precision (mAP), nearest neighbor distance, e-Measure, or first/second tier (FT/ST).

# Bibliography


Abbasi, S., Mokhtarian, F., and Kittler, J. (1999). Curvature scale space image in shape similarity retrieval. *Multimedia Systems*, 7(6):467–476.

Adamek, T. and O'Connor, N. (2004). A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):742–753.

Alajlan, N., Kamel, M., and Freeman, G. (2008). Geometry-based image retrieval in binary image databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1003–1013.

Atabay, H. A. (2017). Binary shape classification using convolutional neural networks. *IIOAB Journal*, 7(October 2016):332–336.

Atienza, R. (2019). Pyramid u-network for skeleton extraction from shape points. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Attalla, E. and Siy, P. (2005). Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching. *Pattern Recognition*, 38(12):2229–2241.

Au, O. K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., and Lee, T.-Y. (2008). Skeleton extraction by mesh contraction. *ACM Trans. Graph.*, 27(3):44:1–44:10.

Aubert, G. and Aujol, J. F. (2014). Poisson skeleton revisited: A new mathematical perspective. *Journal of Mathematical Imaging and Vision*, 48(1):149–159.

Aubry, M., Schlickewei, U., and Cremers, D. (2011). The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE.

Bai, X., Liu, W., Tu, Z., and Angeles, L. (2009). Integrating Contour and Skeleton for Shape Classification. In *Workshop on NORDIA (in conjunction with ICCV09)*.

Bai, X., Yang, X., Latecki, L., Liu, W., and Tu, Z. (2010). Learning context-sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):861–874.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.

Beristain, A. and Grana, M. (2010). Pruning algorithm for voronoi skeletons. *Electronics Letters*, 46(1):39–41.

Bernard, T. M. and Manzanera, A. (1999). Improved low complexity fully parallel thinning algorithm. In *Proceedings 10th International Conference on Image Analysis and Processing*, pages 215–220.

Bhuptani, N. and Talati, B. (2014). Variations in shape context descriptor: A survey. *International Journal of Computer Applications*, 90(12):29–33.

Biasotti, S., Falcidieno, B., Giorgi, D., and Spagnuolo, M. (2014). *Mathematical Tools for Shape Analysis and Description*. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool Publishers.

Blum, H. (1967). A Transformation for Extracting New Descriptors of Shape. In Wathen-Dunn, W., editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge.

Butt, M. A. and Maragos, P. (1998). Optimum design of chamfer distance transforms. *IEEE Transactions on Image Processing*, 7(10):1477–1484.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Chaudhari, A. J., Leahy, R. M., Wise, B. L., Lane, N. E., Badawi, R. D., and Joshi, A. A. (2014). Global point signature for shape analysis of carpal bones. *Physics in Medicine and Biology*, 59(4):961–973.

Chaudhry, R., Ofli, F., Kurillo, G., Bajcsy, R., and Vidal, R. (2013). Bio-inspired dynamic 3d discriminative skeletal features for human action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE.

Chaussard, J., Couprie, M., and Talbot, H. (2011). Robust skeletonization using the discrete $\lambda$-medial axis. *Pattern Recognition Letters*, 32(9):1384–1394.

Chui, C. K., Lin, S.-B., and Zhou, D.-X. (2019). Deep neural networks for rotation-invariance approximation and learning. *ArXiv*, abs/1904.01814.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. (2018). Spherical CNNs. In *International Conference on Learning Representations*.

Couprie, M., Coeurjolly, D., and Zrour, R. (2007). Discrete bisector function and Euclidean skeleton in 2D and 3D. *Image and Vision Computing*, 25(10):1543–1556.

do Carmo, M. (1992). *Riemannian Geometry*. Mathematics (Boston, Mass.). Birkhäuser.

Drew, M. S., Lee, T. K., and Rova, A. (2009). Shape retrieval with eigen-CSS search. *Image and Vision Computing*, 27(6):748–755.

Dubuisson, M.-P. and Jain, A. (2002). A modified Hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 566–568. IEEE Comput. Soc. Press.

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. (2018a). Learning so(3) equivariant representations with spherical cnns. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 54–70, Cham. Springer International Publishing.

Esteves, C., Allen-Blanchette, C., Zhou, X., and Daniilidis, K. (2018b). Polar transformer networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Felzenszwalb, P. F. and Schwartz, J. D. (2007). Hierarchical matching of deformable shapes. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Feng, Y., Zhang, Z., Zhao, X., Ji, R., and Gao, Y. (2018). Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Figueiredo, M. A. T., Leitao, J. M. N., and Jain, A. K. (2000). Unsupervised contour representation and estimation using b-splines and a minimum description length criterion. *IEEE Transactions on Image Processing*, 9(6):1075–1087.

Freifeld, O. and Black, M. J. (2012). Lie Bodies: A Manifold Representation of 3D Human Shape. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *European Conference on Computer Vision*, number October 2012 in Lecture Notes in Computer Science, pages 1–14. Springer International Publishing, Cham.

Gao, F., Wei, G., Xin, S., Gao, S., and Zhou, Y. (2018). 2D skeleton extraction based on heat equation. *Computers and Graphics (Pergamon)*, 74:99–108.

Gao, Z., Yu, Z., and Pang, X. (2014). A compact shape descriptor for triangular surface meshes. *Computer-Aided Design*, 53:62–69.

Giesen, J., Miklos, B., Pauly, M., and Wormser, C. (2009). The scale axis transform. In *Proceedings of the 25th annual symposium on Computational geometry - SCG '09*, page 106, New York, New York, USA. ACM Press.

Gorelick, L., Galun, M., and Sharon, E. (2006). Shape Representation and Classification Using the Poisson Equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1991–2005.

Grigorescu, C. and Petkov, N. (2003). Distance sets for shape filters and shape recognition. *IEEE Transactions on Image Processing*, 12(10):1274–1286.

Hesselink, W. H. and Roerdink, J. B. (2008). Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2204–2217.

Huang, J. and You, S. (2016). Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE.

Kanezaki, A. (2016). Rotationnet: Learning object classification using unsupervised viewpoint estimation. *CoRR*, abs/1603.06208.

Kendall, D. G. (1977). The diffusion of shape. *Advances in Applied Probability*, 9(3):428–430.

Kendall, D. G. (1984a). Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121.

Kendall, D. G. (1984b). Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121.

Khotanzad, A. and Hong, Y. H. (1990). Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497.

Kim, H.-K. and Kim, J.-D. (2000). Region-based shape descriptor invariant to rotation, scale and translation. *Signal Processing: Image Communication*, 16(1-2):87–93.

KING, D. B., WERTHEIMER, M., KELLER, H., and CROCHETIÈRE, K. (1994). The legacy of max wertheimer and gestalt psychology. *Social Research*, 61(4):907–935.

Koffka, K. (1999). *Principles of Gestalt Psychology*. Cognitive psychology]. Routledge.

Kokkinos, I., Bronstein, M. M., Litman, R., and Bronstein, A. M. (2012). Intrinsic shape context descriptors for deformable shapes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 159–166.

Kondor, R. and Trivedi, S. (2018). On the generalization of equivariance and convolution in neural networks to the action of compact groups. *arXiv preprint arXiv:1802.03690*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Kulon, D., Wang, H., Güler, R. A., Bronstein, M., and Zafeiriou, S. P. (2019). Single image 3d hand reconstruction with mesh convolutions. *ArXiv*, abs/1905.01326.

Laga, H. (2018). A survey on nonrigid 3d shape analysis. In *Academic Press Library in Signal Processing, Volume 6*, pages 261–304. Elsevier.

Latecki, L. J. and Lakamper, R. (2000). Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, R. N. (1984). Two-dimensional critical point configuration graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):442–450.

Li, C. and Ben Hamza, A. (2013). A multiresolution descriptor for deformable 3d shape retrieval. *Vis. Comput.*, 29(6-8):513–524.

Li, H., Sun, L., Wu, X., and Cai, Q. (2018). Scale-invariant wave kernel signature for non-rigid 3d shape retrieval. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE.

Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Actional-structural graph convolutional networks for skeleton-based action recognition. *ArXiv*, abs/1904.12659.

Li, R., Bu, G., and Wang, P. (2017). An automatic tree skeleton extracting method based on point cloud of terrestrial laser scanner. *International Journal of Optics*, 2017:1–11.

Limberger, F. A. and Wilson, R. C. (2015). Feature encoding of spectral signatures for 3d non-rigid shape retrieval. In *Procedings of the British Machine Vision Conference 2015*. British Machine Vision Association.

Ling, H., Member, S., and Jacobs, D. W. (2007). Shape Classification Using the Inner-Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299.

Litany, O., Bronstein, A. M., Bronstein, M. M., and Makadia, A. (2018). Deformable shape completion with graph convolutional autoencoders. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1886–1895.

Litman, R. and Bronstein, A. M. (2014). Learning spectral descriptors for deformable shape correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):171–180.

Liu, Y. K. and Žalik, B. (2005). An efficient chain code with huffman coding. *Pattern Recognition*, 38(4):553–557.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Marie, R., Labbani-Igbida, O., and Mouaddib, E. M. (2016). The Delta Medial Axis: A fast and robust algorithm for filtered skeleton extraction. *Pattern Recognition*, 56:26–39.

Masoumi, M., Li, C., and Hamza, A. B. (2016). A spectral graph wavelet approach for nonrigid 3d shape retrieval. *Pattern Recognition Letters*, 83:339–348.

Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928.

McNeill, G. and Vijayakumar, S. (2005). 2d shape classification and retrieval. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1483–1488, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

McNeill, G. and Vijayakumar, S. (2006). Hierarchical procrustes matching for shape retrieval. In *Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*. IEEE.

Miklos, B., Giesen, J., and Pauly, M. (2010). Discrete scale axis representations for 3d geometry. *ACM Trans. Graph.*, 29(4):101:1–101:10.

Mokhtarian, F., Abbasi, S., and Kittler, J. (1998). Efficient and robust retrieval by shape content through curvature scale space. In *Series on Software Engineering and Knowledge Engineering*, pages 51–58. WORLD SCIENTIFIC.

Mokhtarian, F. and Bober, M. (2003). *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization*. Springer Netherlands.

Mokhtarian, F. and Mackworth, A. (1992). A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805. cited By 722.

Mori, G., Belongie, S., and Malik, J. (2005). Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837.

Nava-Yazdani, E., Hege, H.-C., Sullivan, T., and von Tycowicz, C. (2019). Geodesic analysis in kendall's shape space with epidemiological applications. *arXiv preprint arXiv:1906.11950*.

Ogniewicz, R. and Ilg, M. (1992). Voronoi skeletons: theory and applications. In *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 63–69.

Peter, A., Rangarajan, A., and Ho, J. (2008). Shape l'anerouge: Sliding wavelets for indexing and retrieval. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Peters, R. and Ledoux, H. (2016). Robust approximation of the medial axis transform of LiDAR point clouds as a tool for visualisation. *Computers & Geosciences*, 90:123–133.

Pickup, D., Sun, X., Rosin, P. L., and Martin, R. R. (2016a). Skeleton-based canonical forms for non-rigid 3d shape retrieval. *Computational Visual Media*, 2(3):231–243.

Pickup, D., Sun, X., Rosin, P. L., Martin, R. R., Cheng, Z., Lian, Z., Aono, M., Hamza, A. B., Bronstein, A., Bronstein, M., Bu, S., Castellani, U., Cheng, S., Garro, V., Giachetti, A., Godil, A., Isaia, L., Han, J., Johan, H., Lai, L., Li, B., Li, C., Li, H., Litman, R., Liu, X., Liu, Z., Lu, Y., Sun, L., Tam, G., Tatsuma, A., and Ye, J. (2016b). Shape retrieval of non-rigid 3d human models. *International Journal of Computer Vision*, 120(2):169–193.

Postolski, M., Couprie, M., and Janaszewski, M. (2014). Scale filtered Euclidean medial axis and its hierarchy. *Computer Vision and Image Understanding*, 129:89–102.

Pumarola, A., Agudo, A., Porzi, L., Sanfeliu, A., Lepetit, V., and Moreno-Noguer, F. (2018). Geometry-aware network for non-rigid shape prediction from a single view. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4681–4690.

Punam K. Saha, G. B. and de Baja (Eds.), G. S. (2017). *Skeletonization. Theory, Methods and Applications*. Elsevier Science, 1st edition edition.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85.

Qi, C. R., Su, H., Niessner, M., Dai, A., Yan, M., and Guibas, L. J. (2016). Volumetric and multi-view cnns for object classification on 3d data. *arXiv preprint arXiv:1604.03265*.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*.

Qiu, T., Yan, Y., and Lu, G. (2011). A medial axis extraction algorithm for the processing of combustion flame images. In *2011 Sixth International Conference on Image and Graphics*, pages 182–186.

Reuter, M., Wolter, F.-E., and Peinecke, N. (2006). Laplace–beltrami spectra as 'shape-DNA' of surfaces and solids. *Computer-Aided Design*, 38(4):342–366.

Rumpf, M. and Preusser, T. (2002). A level set method for anisotropic geometric diffusion in 3d image processing. *SIAM Journal on Applied Mathematics*, 62(5):1772–1793.

Rustamov, R. M. (2007). Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 225–233, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Safar, M. H. and Shahabi, C. (2003). *Shape Analysis and Retrieval of Multimedia Objects*. Springer US.

Saha, P. K., Borgefors, G., and Sanniti di Baja, G. (2016). A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12.

Sato, M., Bitter, I., Bender, M. A., Kaufman, A. E., and Nakajima, M. (2000). Teasar: tree-structure extraction algorithm for accurate and robust skeletons. In *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*, pages 281–449.

Sebastian, T., Klein, P., and Kimia, B. (2003). On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125.

Sebastian, T. B., Klein, P. N., and Kimia, B. B. (2004). Recognition of shapes by editing shock graphs. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 00(528):755–762.

Shen, W., Wang, X., Yao, C., and Bai, X. (2014a). Shape recognition by combining contour and skeleton into a mid-level representation. In *CCPR*.

Shen, W., Wang, X., Yao, C., and Bai, X. (2014b). Shape recognition by combining contour and skeleton into a mid-level representation. In *Communications in Computer and Information Science*, pages 391–400. Springer Berlin Heidelberg.

Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., and Zucker, S. W. (1999). Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32.

Smeets, D., Fabry, T., Hermans, J., Vandermeulen, D., and Suetens, P. (2009). Isometric deformation modelling for object recognition. In *Computer Analysis of Images and Patterns*, pages 757–765. Springer Berlin Heidelberg.

Sobiecki, A., Jalba, A., and Telea, A. (2014). Comparison of curve and surface skeletonization methods for voxel shapes. *Pattern Recognition Letters*, 47:147–156.

Sobiecki, A., Yasan, H. C., Jalba, A. C., and Telea, A. C. (2013). Qualitative Comparison of Contraction-Based Curve Skeletonization Methods. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7883 LNCS, pages 425–439. Springer.

Stiene, S., Lingemann, K., Nuchter, A., and Hertzberg, J. (2006). Contour-based object detection in range images. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 168–175.

Stoyan, D. (1989). [a survey of the statistical theory of shape]: Comment. *Statist. Sci.*, 4(2):115–116.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 945–953.

Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum*, 28(5):1383–1392.

Super, B. (2004). Learning chance probability functions for shape retrieval or classification. In *Conference on Computer Vision and Pattern Recognition Workshop*. IEEE.

Super, B. J. (2006). RETRIEVAL FROM SHAPE DATABASES USING CHANCE PROBABILITY FUNCTIONS AND FIXED CORRESPONDENCE. *International Journal of Pattern Recognition and Artificial Intelligence*, 20(08):1117–1137.

Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., and Telea, A. (2016). 3d skeletons: A state-of-the-art report. *Computer Graphics Forum*, 35(2):573–597.

Tal, A. (2014). *3D Shape Analysis for Archaeology*, pages 50–63. Springer Berlin Heidelberg, Berlin, Heidelberg.

Thompson, D. W. (1942). *On growth and form / by D'Arcy Wentworth Thompson*. Cambridge University Press Cambridge, Eng, 2nd ed. edition.

Toshev, A. (2011). *Shape Representations For Object Recognition*. PhD thesis, University of Pennsylvania.

Toshev, A., Taskar, B., and Daniilidis, K. (2012). Shape-based object detection via boundary structure segmentation. *International Journal of Computer Vision*, 99(2):123–146.

Tsogkas, S. and Dickinson, S. J. (2017). Amat: Medial axis transform for natural images. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2727–2736.

Tu, Z. and Yuille, A. L. (2004). Shape matching and recognition – using generative models and informative features. In *Lecture Notes in Computer Science*, pages 195–209. Springer Berlin Heidelberg.

van der Maaten, L. J. P., Boon, P. J., Lange, G., Paijmans, H., and Postma, E. O. (2006). Computer vision and machine learning for archaeology. In *Proceedings of the Computer Applications in Archaeology, CAA 2006*, page in press. Dr. H. Kamermans, Faculty of Archeology, Leiden University.

Viswanathan, G. K., Murugesan, A., and Nallaperumal, K. (2013). A parallel thinning algorithm for contour extraction and medial axis transform. In *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, pages 606–610.

Wachinger, C., Salat, D. H., Weiner, M., and and, M. R. (2016). Whole-brain analysis reveals increased neuroanatomical asymmetries in dementia for hippocampus and amygdala. *Brain*, 139(12):3253–3266.

Wafi, N. M., Yaakob, S. N., Salim, N. S., Jusoh, M., Nazren, A. R. A., and Hisham, M. B. (2016). Image analysis using new descriptors average feature optimization based on fourier descriptors technique. In *2016 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, pages 135–138.

Wang, C., Cheng, M., Sohel, F., Bennamoun, M., and Li, J. (2019a). NormalNet: A voxel-based CNN for 3d object classification and retrieval. *Neurocomputing*, 323:139–147.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019b). Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*.

Wang, Y., Xu, Y., Tsogkas, S., Bai, X., Dickinson, S. J., and Siddiqi, K. (2018). Deepflux for skeletons in the wild. *ArXiv*, abs/1811.12608.

Worrall, D. E. and Brostow, G. J. (2018). Cubenet: Equivariance to 3d rotation and translation. *CoRR*, abs/1804.04458.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). Harmonic networks: Deep translation and rotation equivariance. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Xie, J., Heng, P.-A., and Shah, M. (2008). Shape matching and modeling using skeletal context. *Pattern Recognition*, 41(5):1756 – 1767.

Yang, S. and Wang, Y. (2007). Rotation invariant shape contexts based on feature-space fourier transformation. In *Fourth International Conference on Image and Graphics (ICIG 2007)*, pages 575–579.

Yang, X., Koknar-Tezel, S., and Latecki, L. J. (2009). Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Ye, J. and Yu, Y. (2015). A fast modal space transform for robust nonrigid shape retrieval. *The Visual Computer*, 32(5):553–568.

Ye Mei and Androutsos, D. (2008). Affine invariant shape descriptors: The ica-fourier descriptor and the pca-fourier descriptor. In *2008 19th International Conference on Pattern Recognition*, pages 1–4.

Zeng, A., Song, S., Nießner, M., Fisher, M., and Xiao, J. (2016). 3dmatch: Learning the matching of local 3d geometry in range scans. *CoRR*, abs/1603.08182.

Zhang, T. Y. and Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239.

Zhao, Y. and Belkasim, S. (2012). Multiresolution fourier descriptors for multiresolution shape analysis. *IEEE Signal Processing Letters*, 19(10):692–695.

Zhihu Huang and Jinsong Leng (2010). Analysis of hu's moment invariants on image scaling and rotation. In *2010 2nd International Conference on Computer Engineering and Technology*, volume 7, pages V7–476–V7–480.

Zhirong Wu, Song, S., Khosla, A., Fisher Yu, Linguang Zhang, Xiaoou Tang, and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920.