Medial Measures for Recognition, Mapping and Categorization

Morteza Rezanejad

Doctor of Philosophy

School of Computer Science
McGill University
Montreal, Quebec, Canada

August 2019

A thesis submitted to McGill University in partial
fulfillment of the requirements of the degree of
Doctor of Philosophy

Dedication

This thesis is dedicated to:

- Nathan Altshiller Court (1881-1968) who was a Polish-American mathematician, a geometer in particular, and author of the famous book: College Geometry - An Introduction to the Modern Geometry of the Triangle and the Circle [101].
- Maryam Mirzakhani (1977-2017) who was an Iranian mathematician and a professor of mathematics at Stanford University. She was the first and only woman to win the prestigious Fields Medal.

Acknowledgements

My greatest thanks are to my advisor, Professor Kaleem Siddiqi. He made me feel supported all the time and helped me in all areas of my graduate life gently. He has been an exceptional mentor and supervisor to me and I am thankful for having had the opportunity to train under him. I spent my entire 20s studying and growing under his supervision and I can not think of anyone else in my life who had this much influence on me. I am thankful to him for being always there to support me. During my Ph.D. studies, I was fortunate to be able to collaborate with several wonderful professors, postdoctoral fellows, and colleagues. In particular, I am deeply thankful to Professor Sven Dickinson, Professor Allan Jepson, Professor Dirk Bernhardt-Walther, Dr. John Wilder, Professor Gregory Dudek, Professor Ioannis Rekleitis, Professor Guido Gerig, Professor Louis Collins, Professor Hervé Lombaert, Professor Frank Ferrie, Professor Peter Savadjiev, Dr. Diego Alejandro Macrini, Dr. Stavros Tsogkas, Dr. Haz-Edine Assemlal and Mr. Benoit Fiset. Also, I would like to thank my previous and current lab-mates in the Shape analysis group at McGill University, especially, Svetlana Stolpner, Parya Momayyez, Emmanuel Piuze, Chu Wang, Damien Goblot, Tristan Aumentado-Armstrong, Tabish Syed, Amir Kadivar, Pulkit Khandelwal, Aashima Singh, and Gabriel Downs. I want to particularly express my gratitude to two of my closest friends and colleagues at the Centre for Intelligent Machines: Babak Samari and Bahareh Ghotbi. There are many memories of my time together with them that I will always cherish. I am also grateful to several of my old friends who have supported me daily: Pantea Ahmadi, Hassan Mozaffari, Florian Shkurti, Aakash Nandi, Iman Hazrati Ashtiani, Mehrnoosh Abedi, Saman Rahimi Mousavi, Kamran Ahmadpour, Fahim Javid, Negin Vatandoost and many others. You have all been behind me, even through difficult times! A special thank you to Krys Dudek who served as the Program Administrator for the NSERC CREATE Program for Medical Image Analysis and was always there to help me. Finally, I would like to thank my family for being there for me. My greatest gratitude is to my wife, Elham, who stood by my side and provided guidance to me every step of the way. Without her support, this journey through my doctoral years would not have been possible.

Abstract

Visual shape analysis plays a fundamental role in perception by man and by computer, allowing for inferences about properties of objects and scenes in the physical world. Mathematical approaches to describing visual form can benefit from the use of representations that simultaneously capture properties of an object's outline as well as its interior. Motivated by the success of medial models, this doctoral thesis revisits a quantity related to medial axis computations, the average outward flux of the gradient of the Euclidean distance function from a boundary, and then addresses three distinct problems using this measure. First, I consider the problem of view sphere partitioning for view-based object recognition from sparse views. View-based 3D object recognition requires a selection of model object views against which to match a query view. Ideally, for this to be computationally efficient, such a selection should be sparse. To address this problem, I introduce a novel hierarchical partitioning of the view sphere into regions within which the silhouette of a model object is qualitatively unchanged. To achieve this, I propose a part-based abstraction of a skeleton, as a graph, dubbed the Flux Graph, which allows for views to be grouped. Next, I consider the problem of mapping an initially-unknown 2D environment from possibly noisy sensed samples via an on-line procedure which robustly computes a retraction of its boundaries to obtain a topological representation. Here I devise an algorithm that allows for online map construction with loop closure. I demonstrate that the proposed method allows the robot to localize itself on a partially constructed map to calculate a path to unexplored parts of the environment (frontiers), to compute a robust

terminating condition when the robot has fully explored the environment, and finally to achieve loop closure detection. I also show that the resulting map is stable under perturbations to the sensed boundary, and to variations in starting locations for exploration. Finally, I consider the problem of scene categorization from complex line drawings. In the context of human vision, we show that local ribbon symmetry between neighboring pairs of contours facilitates the categorization of complex real-world environments by human observers. In the context of computer vision, I demonstrate a high level of performance in the problem of convolutional neural network-based recognition of natural scenes from line drawings, even in the absence of color, texture and shading information. I then show that the inclusion of medial-axis based contour salience weights leads to a further boost in recognition performance, adding useful information that does not appear to be exploited when the neural networks are trained on contours alone.

Abrégé

L'analyse de la forme visuelle joue un rôle fondamental dans la perception par l'humain et par l'ordinateur, qui permet de déduire des propriétés d'objets et de scènes du monde physique. Les approches mathématiques pour décrire la forme visuelle peuvent bénéficier de l'utilisation de représentations qui capturent simultanément les propriétés du contour d'un objet ainsi que son intérieur. Motivée par le succès des modèles médiaux, ma thèse de doctorat revisite les calculs liés à l'axe médian et aux flux sortants moyens du gradient de la fonction de distance euclidienne à partir d'une frontière, puis propose des solutions à trois problèmes distincts en utilisant les résultats. En premier lieu, je considère le problème du partitionnement d'un vue sphérique pour la reconnaissance d'objet basée sur des vues fragmentées. La reconnaissance d'objet 3D basée sur un modèle, nécessite une recherche dans une base de collection, d'un modèle de l'objet de la requête de comparaison. Idéalement, pour que cela soit efficace sur le plan informatique, une telle sélection devrait être éparse. Pour résoudre ce problème, j'introduis un nouveau partitionnement hiérarchique de la vue sphérique en régions dans lesquelles la silhouette d'un objet modèle reste qualitativement inchangée. Pour ce faire, je propose une abstraction basée sur les parties du squelette, sous forme de graph, appelé « Flux Graph », qui permet de regrouper les vues. Ensuite, je considère le problème de la cartographie d'un environnement 2D initialement inconnu, à partir d'échantillons images ayant du bruit, via une procédure en temps réel qui calcule de manière robuste, une rétraction de ses limites pour en extraire une représentation topologique. Ici, j'ai conçu un algorithme qui permet la construction

de cartes en temps réel avec une fermeture de la boucle. Je démontre que la méthode proposée permet au robot de se localiser sur une carte partiellement construite afin de calculer un chemin vers des parties inexplorées de l'environnement (frontières) afin de calculer une condition de fin robuste lorsque le robot a entièrement exploré l'environnement, puis de réaliser une détection de fermeture de la boucle. Je démontre également que la carte obtenue est stable en cas de perturbation des frontières détectées et des lieux de départ de l'exploration. Enfin, je considère le problème de la catégorisation des scènes à partir de dessins au trait complexes. Dans le contexte de la vision humaine, nous montrons que la symétrie de lignes parallèles (rubans) entre des paires de contours voisins facilite la catégorisation d'environnements complexes du monde réel par les observateurs humains. Dans le contexte de la vision par ordinateur, je démontre un haut niveau de performance dans le problème de la reconnaissance convolutive basée sur un réseau neuronal de dessins au trait de scènes de la nature, même en l'absence d'informations de couleur, de texture et d'ombrage. Je montre ensuite que l'inclusion de poids de saillance des contours basés sur l'axe médian conduit à une amélioration supplémentaire des performances de reconnaissance des scènes de la nature, en ajoutant des informations utiles qui ne semblent pas être exploitées lorsque les réseaux de neurones sont formés uniquement sur les contours.

Acronyms

AOF	Average Outward Flux
CNN	Convolutional Neural Network
CPD	Coherent Point Drift
CPU	Central Processing Unit
DAG	Directed Acyclic Graph
FOCUSR	Feature-Oriented Correspondence Using Spectral Regularization
GVG	Generalized Voronoi Graph
MAT	Medial Axis Transform
RAM	Random Access Memory
RANSAC	Random Sample Consensus
RGB	Red, Green, Blue
ROS	Robot Operating System
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SLS	Smoothed Local Symmetry
SVM	Support Vector Machine
TSV	Topological Signature Vector
VGG	Visual Geometry Group

Symbols

$\mathbf{T}$	Tangent vector
$\mathbf{N}$	Normal vector
r	Medial axis radius function
$\mathbf{b}^{\pm\mathbf{1}}$	Bi-tangent points
θ	Object angle
$\mathbf{p}$	Point
d	Distance function
$\dot{\mathbf{q}}$	Gradient of the Euclidean distance function
$Sk(\Omega)$	The Blum interior medial locus or skeleton of Ω
G	Graph
Γ	Topological similarity
Δ	Geometric similarity

Contents

List of Figures

List of Tables

Part I

Introduction, Background and Geometry of the Medial Axis

1

Introduction

From trying to draw the plan of our house at a young age to attempting more difficult geometric problems in my high school years, I have always been fascinated by our visual perception of objects and scenes in this world. Possibly the most influential book I have ever read is Nathan Altshiller Court's College Geometry - An Introduction to the Modern Geometry of the Triangle and the Circle [101]. Growing up in the computer age has since motivated me to work on machine algorithms to solve geometric problems, as humans can do. What you read here is a small effort in that direction.

Whereas present computer vision systems are competitive, they fail in situations where humans succeed, and they are extremely data-hungry. As an example, while a computer vision system may need to train on hundreds of images of a cat, a child can learn this ab-

stract category from a few example sketches of their outlines. Observations such as this, highlight a role for visual shape analysis in computer vision-based systems, for abstraction, efficiency, robustness, and generalization.

Visual shape analysis plays a fundamental role in perception by man and by computer and allows for inferences about properties of objects and scenes in the physical world. Although the problem of form analysis is not always mathematically well defined, researchers have tackled it by dividing it into more specific tasks. Once these tasks are solved, the solutions can be put together to address more complex visual shape analysis problems that arise in many important applications such as in robotics, biology, medicine, and industry. It is no surprise that the range of applications can be very broad for this topic, considering human vision capabilities. The reader is referred to Biederman [10], Edelman [43], Leyton [84], Perrett and Oram [107], Kimia et al. [74] for further background on this subject.

Of particular interest to the focus of this thesis are the problems of recognition or categorization in computer and human vision, and environment mapping in robotics. Humans can rapidly spot a wide range of things when they look at an image or watch a video. Being able to detect and differentiate between people, object or scene categories is one of the capabilities often sought. Advances in developing techniques that enable machines to tackle such problems have played a prominent role in modern-day computer vision. Looking at a known 3D object, humans can readily recognize both the object and the vantage point from which it is viewed. For machines, one of the complexities of object recognition is to deal with multiple views of 3D objects when projected in 2D. A 3D object can be seen

from an infinite number of viewpoints, but not all of them necessarily provide drastically different perceptual information. For humans, the viewpoints can be interpreted in simple terms most of the time, e.g., one can easily recognize the different views a car (top, front, back, bottom, or sides). For machines, this ability is not instant, in fact, it is one of the very challenging problems in the area of computer vision. This thesis explores a method that could lead to a better understanding of view-based recognition problems. The robotics community has also benefited from the application of visual shape analysis techniques in different problems related to navigation and mapping. Being able to design robots that can sense an environment autonomously and navigate through it safely is of key importance in this field. Having environments mapped as visual shapes that could be interpreted by a robot as a set of nodes and edges, this thesis uses a shared representation with that used in view-based recognition problems and extends it to devise a proof of concept application for autonomous environment mapping problems. Shapes can be thought of as words of a visual language [33], and as a result, a visual scene can be viewed as a collection of regions and the boundaries surrounding them. Taking this interpretation, this thesis interprets and analyzes scenes in terms of these visual language words specifically, contours in scenes, applying the same representation used in addressing the previous problems.

To be able to tackle these problems of object or scene categorization, or environment mapping computationally, it is helpful to have a versatile representation for object shape. In this thesis we will often refer to the projected contours of objects, onto an image plane, but also to the more general layout of bounding contours in a scene (see Figure 1.1).

One may ask what features a reliable shape representation method should have. An

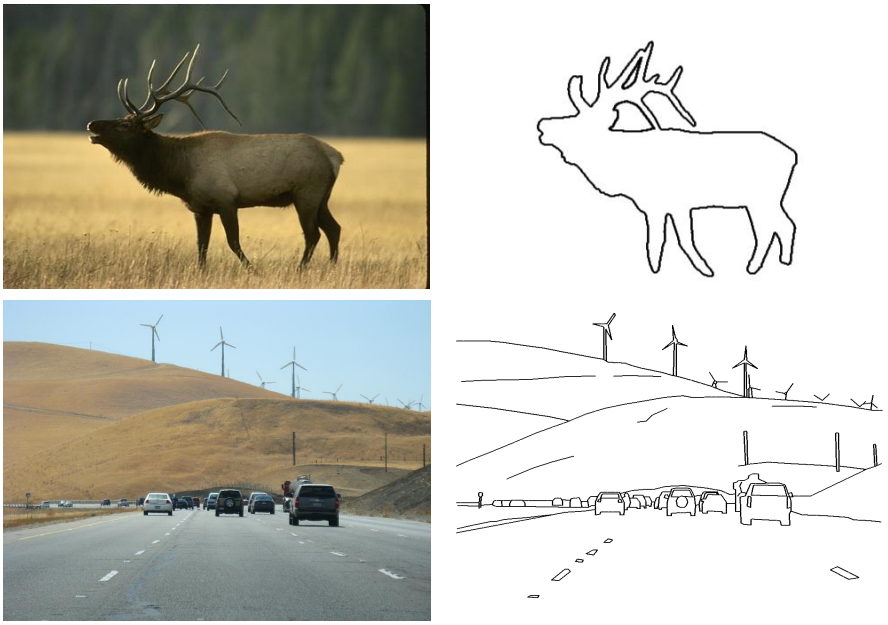


Figure 1.1: This figure shows examples of photographs versus silhouettes or line drawings of scenes. Human observers can just as easily recognize the objects and the underlying scenes from the projected outlines as they can from the photographs.

ability to cover a large class of exemplars with a unified representation is often desirable. Besides, it should be possible to extract parts and subparts from the shape representation. In many applications, the shape model needs to deform to fit a different setting or a different exemplar, which requires the description to preserve its fundamental elements in the presence of some deformations. A shape representation method can be judged by several criteria, including:

1. Completeness: It should apply to a large class of examples.
2. Hierarchy: Hierarchical relationships between visual entities including parts and their sub-parts should be recoverable.

3. Invariance: The representation should allow for common transformations including Euclidean ones such as rotation and translation, as well as modest changes due to scaling.
4. Stability: In that, the representation should be able to handle boundary deformations that do not change the qualitative appearance of outline shape.
5. Similarity: In that, the representation should allow for measuring the degree to which two exemplars are similar.

Contour based representations of object outlines are a popular choice for problems of categorization and recognition. Such approaches use boundary information to extract salient features. Representations that are sensitive to boundary details can suffer from being unstable in the presence of boundary perturbations in the absence of an appropriate notion of scale. Hence, defining the notion of a shape part is difficult with representations that exploit just boundary data. In the region-based approaches, the interior of an object is taken into account, which eases the task of describing underlying object parts.

Some information is lost when a 3D real-world object or scene is projected onto an image plane. This makes the task of finding a robust image-based description of its projected outline, challenging. Medial representations have served as a popular choice for this task because they take and combine assets from both contour-based and region-based approaches. Blum [15] introduced the notion of medial loci for representing the projected outlines of 3D objects in 2D images. He later suggested an extension of medial loci to the objects in 3D themselves, which were later generalized to skeletons [18, 16, 17]. After that, mathematicians and computer scientists developed these ideas and further extended

them to describe shapes in 2D and 3D images.

The medial axis's power in representing both object interiors and object boundaries provides a unique strength for approaches that require the analysis of geometrical information of visual cues. The medial axis transform (MAT) provides simplified shape representations that have several applications in shape analysis including shape matching, computer animation, topology mapping, and path planning [133].

Motivated by the success of medial representations, my doctoral thesis revisits a quantity related to medial axis computations, the average outward flux (AOF) through a shrinking disk of the gradient of the Euclidean distance function to the boundary of a 2D object [38, 135]. Broadly speaking, I investigate three distinct problems of interest to the computer and human vision and robotics communities:

1. View-based object recognition from sparse views
2. The online abstraction of topological maps for 2D environments, and
3. Scene categorization from line drawings of natural images in both human and computer vision.

I tackle aspects of these problems using visual shape analysis, and for all three I use a shared shape representation which is based on medial axes computed using a notion of average outward flux.

1.1 Objectives of this Thesis

There are three main objectives of this thesis:

1. To address view-based object recognition, with a focus on silhouette shape. To this end I:
    - Propose a simplified yet powerful silhouette representation that can be used to disambiguate distinct views of 3D objects from each other, following projection onto an image plane.
    - Design an effective strategy to learn suitable model views from images taken of a 3D object and to select the best representative candidate views so that a 3D model object can be efficiently captured by a set of selective views.
2. To examine the role of contour geometry in scene perception and categorization tasks, with a focus on:
    - The development of algorithms for detecting relational perceptual measures between contours of line drawings, including symmetry, parallelism, and proximity.
    - An examination of the efficacy of the developed measures for both human observers and convolutional neural networks, by employing both in real-world scene categorization tasks from line drawings of natural images.
    - The creation of machine-generated line drawings of scenes from photographic images to be able to train networks to use these line drawings, along with

derived features, in scene recognition tasks.

3. To address the problem of effective online 2D environment mapping, where I develop and evaluate a strategy for incremental construction of the Generalized Voronoi Graph (GVG) using average outward flux-based skeletons.

1.2 Publications Arising from this Thesis

View Sphere Partitioning and Flux Graphs

1. **(Book Chapter)** M. Rezanejad and K. Siddiqi. Flux graphs for 2D shape analysis. Chapter 3 in Shape Perception in Human and Computer Vision: An Interdisciplinary Perspective. Editors: Sven Dickinson and Zygmunt Pizlo, Springer, 2013.
2. **(Journal)** M. Rezanejad and K. Siddiqi. View Sphere Partitioning via Flux Graphs Boosts Recognition from Sparse Views. Frontiers in ICT: Computer Image Analysis, 2 (2015) 24.

   M. Rezanejad: Was the primary contributor to these articles.

   K. Siddiqi: Collaborated on methodological development, algorithm design, and writing of the manuscripts.

AOF Skeletons for Environment Mapping

3. **(Conference)** M. Rezanejad, B. Samari, I. Rekleitis, K. Siddiqi and G. Dudek. Robust environment mapping using flux skeletons. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5700-–5705.

M. Rezanejad: Was the primary contributor to this article.

B. Samari: Assisted in algorithm design, experiments and writing.

I. Rekleitis, K. Siddiqi and G. Dudek: Collaborated on methodological development, algorithm design, and writing of the manuscript.

Contour-Based Scene Categorization by Human Observers

4. **(Journal)** J. Wilder, M. Rezanejad, S. Dickinson, K. Siddiqi, A. Jepson, and D. Bernhardt-Walther. Local contour symmetry facilitates scene categorization. Cognition 182 (2019): 307-317.
5. **(Conference)** J. Wilder, M. Rezanejad, K. Siddiqi, A. Jepson, S. Dickinson, and D. Bernhardt-Walther. Local contour symmetry facilitates the neural representation of scene categories in the PPA. Conference on Cognitive Computational Neuroscience, Berlin, Germany, September 2019.
6. **(Abstract)** J. Wilder, M. Rezanejad, K. Siddiqi, A. Jepson, S. Dickinson, and D. Bernhardt-Walther. The neural basis of local contour symmetry in scene perception. Vision Science Society, St. Pete Beach, United States, 2019.
7. **(Abstract)** J. Wilder, M. Rezanejad, S. Dickinson, A. Jepson, K. Siddiqi, and D. Bernhardt-Walther. The Perceptual Advantage of Symmetry for Scene Perception. In Journal of Vision, 17 (2017) 1091–1091.
8. **(Abstract)** J. Wilder, M. Rezanejad, S. Dickinson, A. Jepson, K. Siddiqi, and D. Bernhardt-Walther. The role of symmetry in scene categorization by human observers. In Computational and Mathematical Models in Vision (MODVIS), St. Pete

Beach, United States, 2017.

J. Wilder: Led the psychophysics aspect of the work, developed the details of the human experiments and the theory, carried out all the human experiments, and did the majority of writing for these papers.

M. Rezanejad: Developed the AOF medial axes, scores and algorithms for scene symmetry, and also was primarily responsible for the preparation of the splits. Contributed to the analysis along with many other aspects of this research project.

S. Dickinson, K. Siddiqi, A. Jepson, and D. Bernhardt-Walther: collaborated on methodological development, algorithm development, interpretation of the results and writing.

Medial Axis Based Salience Measures for Scene Categorization

9. **(Conference)** M. Rezanejad, G. Downs, J. Wilder, D. Bernhardt-Walther, A. Jepson, S. Dickinson, and K. Siddiqi. Medial Axis Based Contour Salience for Scene Categorization. Computer Vision and Pattern Recognition (CVPR) Conference, June 2019.
10. **(Conference)** M. Rezanejad, G. Downs, J. Wilder, D. Bernhardt-Walther, A. Jepson, S. Dickinson, and K. Siddiqi. Gestalt-based Contour Weights Improve Scene Categorization by CNNs. Conference on Cognitive Computational Neuroscience, Berlin, Germany, September 2019.
11. **(Abstract)** M. Rezanejad, G. Downs, J. Wilder, D. Bernhardt-Walther, S. Dickinson, A. Jepson, and K. Siddiqi. Perceptual grouping aids recognition of line drawings of scenes by CNNs. Vision Science Society, St. Pete Beach, United States,

2019.

12. **(Abstract)** J. Wilder^{+}, M. Rezanejad^{+}, K. Siddiqi, S. Dickinson, A. Jepson, and D. Bernhardt-Walther. Measuring Local Symmetry in Real-World Scenes. In Journal of Vision, 18 (2018) 749-749.

    $^{+}$: Equal contribution.

13. **(Abstract)** M. Rezanejad, J. Wilder, K. Siddiqi, S. Dickinson, A. Jepson, and D. Bernhardt-Walther. Measuring Local Symmetry in Real-World Scenes Using Derivatives of the Medial Axis Radius Function. In Computational and Mathematical Models in Vision (MODVIS), St. Pete Beach, United States, 2018.

14. **(Abstract)** M. Rezanejad, J. Wilder, S. Dickinson, A. Jepson, D. Bernhardt-Walther and K. Siddiqi. Scoring Scene Symmetry. In Computational and Mathematical Models in Vision (MODVIS), St. Pete Beach, United States, 2017.

    M. Rezanejad: Led the computer vision aspects of this work, developed the details of the algorithms and the theory, carried out most of the machine vision experiments, and did the majority of writing for these papers.

    G. Downs: assisted with algorithm implementation, experimental design and in carrying out the CNN experiments.

    J. Wilder: developed the details of the human experiments and helped with writing.

    S. Dickinson, K. Siddiqi, A. Jepson, and D. Bernhardt-Walther: collaborated on methodological development, algorithm development, interpretation of the results and writing.

1.3 Additional Contributions

In addition to the papers I have published with my co-authors on work arising from this thesis, my contributions include:

1. Efficient implementations for computing 2D and 3D AOF skeletons, together with code releases on GitHub:
   - 2D implementation:
     https://github.com/mrezanejad/AOFSkeletons
   - 3D implementation::
     https://github.com/mrezanejad/3DAOFSkeletons
2. The introduction of novel measures for skeletal branch simplification based on AOF skeletons. The simplified skeletons are used to derive a directed graph-based representation which we have termed the "Flux Graph". Open-source implementations are available at https://github.com/mrezanejad/IROS2015.
3. The development of an algorithm to partition the view sphere around an object into regions within which its silhouette is qualitatively unchanged. We show that hierarchical view sphere partitioning boosts 3D object recognition performance in the scenario of matching against a sparse number of model views.
4. The introduction of a novel computational approach, based on the medial axis transform, for measuring the degree of local ribbon symmetry in a line drawing. We show that humans are better in categorizing scenes when shown the 50% more symmetric

pixels and the 50% less symmetric ones.

5. The design of Gestalt-based contour weights, together with a demonstration that such weights improve scene categorization by CNNs. We show that medial-axis based contour salience methods can be used to select more informative subsets of contour pixels and that the variation in CNN classification performance on various choices for these subsets is qualitatively similar to that observed in human performance. Open-source implementations are available at the following GitHub resources:

   - https://github.com/mrezanejad/LineDrawingExtraction
   - https://github.com/mrezanejad/DollarLineDrawing
   - https://github.com/mrezanejad/SalienceScoresForScene.

1.4 Organization of this Thesis

This thesis is organized into three parts. Part I includes the Introduction and Background chapters. Part II includes chapters that use flux graphs, for view sphere partitioning and environment mapping. Finally, Part III includes work on scene categorization from line drawings, both by human observers and CNN based systems. In more detail, Chapter 2 reviews the essential articles in the literature related to the work presented in this thesis. We also provide background on the geometry of the medial axis in 2D in Section 2.1. Chapter 3 introduces a new shape representation approach based on AOF skeletons, which we have termed the flux graph, along with an algorithm for view sphere partitioning using this representation. Chapter 4 describes an online topology mapping algorithm, developed

1.4 Organization of this Thesis

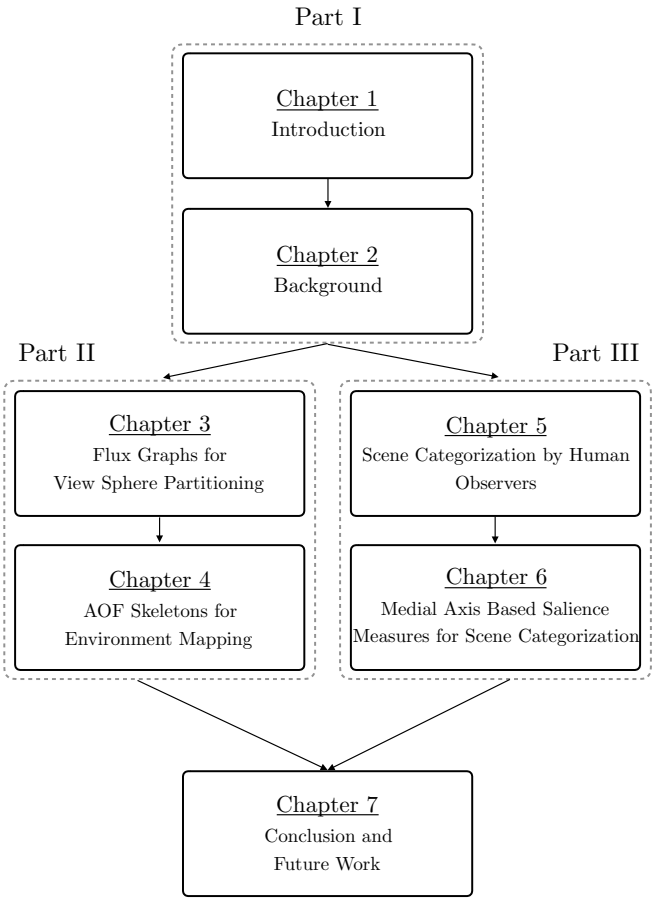


Figure 1.2: Chapters that compose this thesis.

using AOF skeletons, and demonstrates its application to environment mapping. Chapter 5 presents a novel computational approach, based on AOF skeletons, for measuring the degree of local ribbon symmetry in a line drawing and uses this measure in a scene categorization task by human observers. In Chapter 6, we introduce three novel measures of local contour symmetry and local contour separation using AOF skeletons. We show that CNN-based scene categorization systems, just like human observers, can benefit from

explicitly computed contour salience measures derived from such Gestalt grouping cues. Finally, Chapter 7 reviews contributions of this thesis and its shortcomings and suggests directions for future work. For most readers, we recommend reading the chapters in the order prescribed in the flow chart in Figure 1.2.

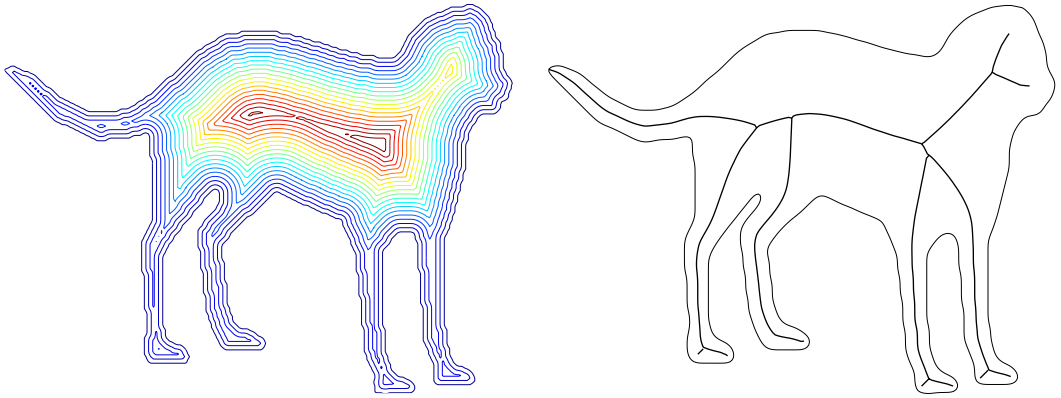
2

Background

In this chapter, we review the material that is relevant to the problems investigated in this thesis. We begin with an overview of the medial axis and average outward flux-based skeletons in Section 2.1. Section 2.2 discusses skeletal graph-based object matching approaches, followed by a brief discussion of other shape and appearance matching methods. Section 2.3 discusses aspect graphs and their role in view-based recognition. Section 2.4 reviews the robotics literature on environment mapping. Finally, Section 2.5 discusses the role of contour geometry in scene categorization. The reader familiar with these topics may choose to move directly to Chapter 3.

2.1 Medial Representations

In Blum's grassfire analogy the medial axis is associated with the quench points of a fire that is lit at the boundary of a field of grass [18]. An equivalent notion of the medial axis is that of the locus of centres of maximal inscribed disks in the region enclosed within a boundary, along with the radii of these disks. The geometry and methods for computing the medial axis that we leverage are based on a notion of average outward flux, as discussed in further detail in Section 2.1.1. Within an object, the loci of medial points form curves about which its outline is locally mirror symmetric. Each point in the interior of the object is the result of the collision of two distinct boundary points, under the action of the grassfire flow. The application of the grassfire process to reveal its quench points along with their radius values is called the *Medial Axis Transform* (MAT). Since the grassfire process is applicable to all bounded shapes, as well as the regions outside of closed shapes, the MAT gives a comprehensive representation in visual shape problems. The medial axis thus consists of the set of points lying inside the boundary that are equidistant to two more points on its boundary. In the following subsection, we will discuss the geometry of the medial axis of an object and we will introduce some notation for the medial representation which will be used frequently later in this thesis (see Figure 2.1).



(a) (b)

Figure 2.1: (a): Iterations of the grassfire process. (b): The resulting skeleton, computed using an average outward flux based method [15].

2.1.1 Geometry of the Medial Axis and Average Outward Flux Skeletons

We begin by formalizing the notion of a skeleton.

Definition 1 *Assume an n-dimensional open object Ω, with its boundary given by $\partial\Omega \in \mathbb{R}^n$ such that $\bar{\Omega} = \Omega \cup \partial\Omega$. A closed disk $D \in \mathbb{R}^n$ is a maximal inscribed disk in $\bar{\Omega}$ if $D \subseteq \bar{\Omega}$ but for any disk D' such that $D \subset D'$, the relationship $D' \subseteq \bar{\Omega}$ does not hold.*

Definition 2 *The Blum interior medial locus or skeleton, denoted by $Sk(\Omega)$, is the locus of centers of all maximal inscribed disks in $\partial\Omega$.*

The term *Skeletal point* is used to refer to a point on the skeleton of an object characterized by a location $\mathbf{p}$ associated with radius of the maximal disk r, object angle θ, direction of the unit tangent vector $\mathbf{T}$, and corresponding boundary points $\mathbf{b}^{\pm\mathbf{1}}$ at that

point, where

$$\theta = \arccos\left(-\frac{dr}{ds}\right) \tag{2.1}$$

with s being the arc length along the medial curve (see Figure 2.2). The points $\mathbf{b}^{\pm\mathbf{1}}$ are referred to as the bi-tangent points.

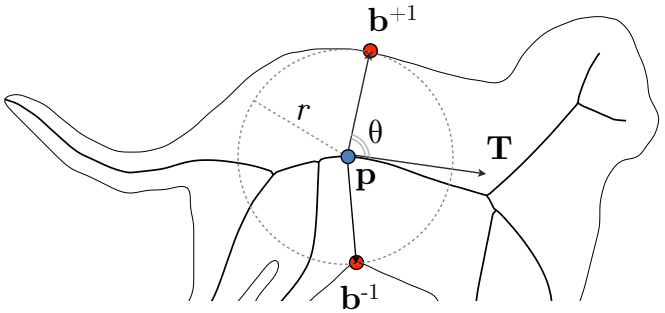


Figure 2.2: Local geometry of a maximal inscribed disk centered at the skeletal point $\mathbf{p}$ with radius r and the object angle θ. The maximal inscribed disk touches the boundary at two points $\mathbf{b}^{\pm\mathbf{1}}$ (adapted from [133]).

Topologically $Sk(\Omega)$ consists of a set of branches that join at branch points to form the complete skeleton. A skeletal branch is a set of contiguous regular points from the skeleton that lie between a pair of junction points, a pair of end points or an end point and a junction point. As shown by Dimitrov et al. [38] these three classes of points can be analyzed by considering the behavior of the average outward flux (AOF) of the gradient of the Euclidean distance function to the boundary of a 2D object through a shrinking disk, where $\dot{\mathbf{q}} = \nabla\mathbf{D}$ [38], with $\mathbf{D}$ the Euclidean distance function to the object's boundary. In the following, we review this computation.

The Euclidean distance between two n-dimensional points $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n)$ and $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_n)$ is the length of the line segment that connects these two points,

and the *Euclidean metric* $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is a function that represents this distance: $d(\mathbf{p}, \mathbf{m}) = \sqrt{\sum_{r=1}^{n} (\mathbf{m}_i - \mathbf{p}_i)^2}$. For each point $\mathbf{p}$, and a given object Ω, a distance metric, can be defined as follows $d_\Omega(\mathbf{p}) = \inf_{\mathbf{m} \in \partial\Omega} d(\mathbf{p}, \mathbf{m})$. The *distance transform* D_Ω of an object Ω is a signed distance function that specifies how close a given point $\mathbf{p}$ is to the boundary of that object $\partial\Omega$. Formally, $\mathcal{D}_\Omega(\mathbf{p}) = d_\Omega(\mathbf{p})$ if $\mathbf{p}$ is inside Ω, $\mathcal{D}_\Omega(\mathbf{p}) = 0$ if $\mathbf{p} \in \partial\Omega$ and $\mathcal{D}_\Omega(\mathbf{p}) = -d_\Omega(\mathbf{p})$ if $\mathbf{p}$ is outside Ω. In Chapters 5 and 6, we consider the interior of regions bounded by contours and in such settings we can assume an unsigned distance function for each region.

Let us define the projection $\Pi(\mathbf{p})$ as the set of closest points on the boundary $\partial\Omega$ to $\mathbf{p}$, i.e., $\Pi(\mathbf{p}) \triangleq \{\mathbf{p}' \in \partial\Omega \ : \ \|\mathbf{p}-\mathbf{p}'\| = \min\{\|\mathbf{p}-\mathbf{p}'\| \forall \ \mathbf{p}' \in \partial\Omega\}\}$. Assume that on the boundary $\partial\Omega$, there exists only one point $\mathbf{m}$ of minimum distance to $\mathbf{p}$, such that $\Pi_\Omega(\mathbf{p}) = \{\mathbf{m}\}$. We then define the *distance function gradient vector* for point $\mathbf{p}$ as: $\dot{\mathbf{q}}_\Omega(\mathbf{p}) = \frac{\mathbf{m}-\mathbf{p}}{||\mathbf{m}-\mathbf{p}||}$. In the case of $|\Pi_\Omega(\mathbf{p})| > 1$, one cannot define the closest boundary point uniquely, and therefore the distance function gradient vector is multivalued. Except for at skeletal points, $\dot{\mathbf{q}}$ is continuous everywhere on its domain and it satisfies the equation: $|\dot{\mathbf{q}}| = 1$. Exploiting the relationship of the integral of the divergence of a vector field within a simply-connected region to the outward flux of that vector field through the boundary of that region leads to a characterization of skeletal points. Let R be a region where its boundary ∂R is a simple closed curve, and $\mathbf{N}$ be the outward normal at each point on the boundary ∂R [38].

Definition 3 *The outward flux of* $\dot{\mathbf{q}}$ *through* ∂R *is defined as*

$$\mathbf{F}(\mathbf{p}) = \int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds.$$

Definition 4 *The average outward flux of* $\dot{\mathbf{q}}$ *through* ∂R *is defined as*

$$AOF = \frac{\int_{\partial R} \langle \dot{\mathbf{q}}, \mathbf{N} \rangle ds}{\int_{\partial R} ds}.$$

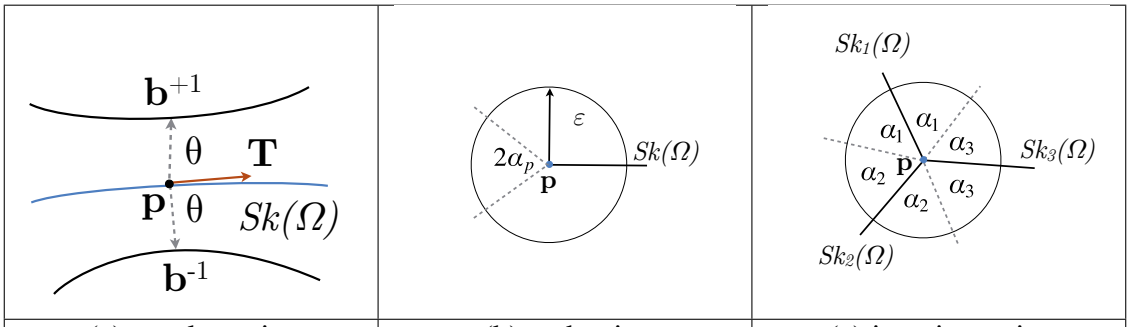
Using the divergence theorem, Dimitrov et al. [38] categorized points into classes by considering the behavior of the average outward flux (AOF) of the gradient of the Euclidean distance function to the boundary of a 2D object, through a shrinking disk. In particular, the limiting AOF value of all points not located on the skeleton is equal to zero.

As discussed in [133], there are three classes of generic skeletal points, whereby generic we mean stable under arbitrarily small perturbations of the boundary. These are regular skeletal points, end points, and junction points. Assume that $\mathbf{F}_\epsilon(\mathbf{p})$ represents the outward flux of a skeletal point $\mathbf{p}$ through a shrinking disk with radius ϵ. Then:

1. $\mathbf{p}$ is a *regular point* if the maximal inscribed disk at $\mathbf{p}$ touches the boundary at two corresponding boundary points, such that $|\Pi_\Omega(\mathbf{p})| = 2$. The computed AOF at a regular point $\mathbf{p}$ is given by $\lim_{\epsilon \to 0} \frac{\mathbf{F}_\epsilon(\mathbf{p})}{2\pi\epsilon} = -\frac{2}{\pi} \sin\theta$.
2. $\mathbf{p}$ is an *end point* if there exists δ, with $(0 < \delta < r)$ such that for any ϵ $(0 < \epsilon < \delta)$ the circle centered at $\mathbf{p}$ with radius ϵ intersects $Sk(\Omega)$ just at a single point (r is the radius of the maximal inscribed disk at $\mathbf{p}$). The computed AOF at an end point $\mathbf{p}$ is given by $\lim_{\epsilon \to 0} \frac{\mathbf{F}_\epsilon(\mathbf{p})}{2\pi\epsilon} = -\frac{1}{\pi}(\sin\theta_{\mathbf{p}} - \theta_{\mathbf{p}})$.
3. $\mathbf{p}$ is a *junction point* if $\Pi_\Omega(\mathbf{p})$ has three or more corresponding closest boundary points. Generically a junction point has degree 3. Branch points of degree higher than 3 are non-generic. The computed AOF at a junction point $\mathbf{p}$ is given

by $\lim_{\epsilon \to 0} \frac{\mathbf{F}_\epsilon(\mathbf{p})}{2\pi\epsilon} = -\frac{1}{\pi} \sum_{i=1}^{n} \sin\theta_i$.

These different classes of skeletal points are shown in Figure 2.3.



(a) regular point | (b) end point | (c) junction point

Figure 2.3: The three classes of skeletal points, shown for segments of the skeleton $Sk(\Omega)$ of a given shape Ω (adapted from [38]).

In this thesis, we shall compute the skeleton by using the following numerical estimate of the limiting AOF around each point $\mathbf{p}$, and then locating those points with non-zero AOF values.

$$AOF = \frac{\int \langle \dot{\mathbf{q}}_{\tilde{\mathbf{p}}}, \mathbf{N}(s) \rangle ds}{2\pi r}, \tag{2.2}$$

where $\tilde{\mathbf{p}} = \{\mathbf{p} + r\mathbf{N}(s)\}$, and $\dot{\mathbf{q}}$ is the distance function at point $\tilde{\mathbf{p}}$. By discretizing the circular area's boundary into n equal arcs, the numerator is approximated by:

$$\int \langle \dot{\mathbf{q}}_{\tilde{\mathbf{p}}}, \mathbf{N}(s) \rangle ds = \frac{2\pi r}{n} \sum_{k=0}^{n-1} \langle \dot{\mathbf{q}}_{\tilde{\mathbf{p}}}, \mathbf{N}(k) \rangle, \tag{2.3}$$

where ds is approximated by division of the perimeter $2\pi r$ by n ($\frac{2\pi r}{n}$), and $\mathbf{N}(k) = (\cos(\frac{2\pi k}{n}), \sin(\frac{2\pi k}{n}))$.

2.1.2 The Shock Graph

Siddiqi et al. [136] classified skeletal points into different categories according to their evolutionary appearance as singularities or shocks of the grassfire flow. These four levels of shocks, originally introduced in [74], are defined below.

1. First-Order (Protrusion): A skeletal point is considered to be a first-order shock if the medial axis radius function within a neighborhood around that skeletal point changes monotonically.
2. Second-Order (Neck): A skeletal point is considered to be a second-order shock if the medial axis radius function is a strict local minimum at it.
3. Third-Order (Bend): A skeletal point is considered to be a third-order shock if the medial axis radius function within a neighborhood around that skeletal point does not change.
4. Fourth-Order (Seed): A skeletal point is considered to be a fourth-order shock if the medial axis radius function is a strict local maximum at it.

Figure 2.4 shows examples of the four different shock types.

Through a merging process, adjacent skeletal points of the same shock order are then merged to make a node. Each node then is labeled based on the shock type of its skeletal points. These nodes then comprise the set of vertices for the shock graph. For the set of edges, connectivity between nodes is determined according to the order of shocks and their topological structure in the skeleton [136]. Existing loops (without considering di-

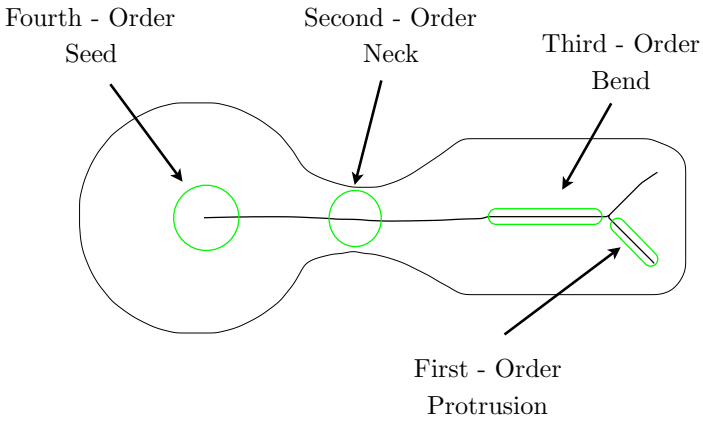


Figure 2.4: Different shock types shown on an arbitrary object. (Adapted from Rezanejad [117])

.

rections over edges) in the shock graphs are eliminated by duplicating tips of loops. A related but slightly different notion of the shock graph is used in [127], where the skeletal points are associated with edges.

2.2 Object Recognition and Matching

Shape matching is the process of taking two or more shapes and comparing them with one another. The evaluation of the quality of the match is typically is based on a similarity or distance measure, computed by comparing primitives representing these shapes. In the following section, we review some of the shape matching approaches in the literature divided into two categories: a) Skeletal graph-based methods b) Non-skeletal graph-based methods.

2.2.1 Skeletal Graph-Based Shape Matching

There are a number of effective shape matching methods in the literature which rely on algorithms to compare the underlying skeletal graphs. In this subsection, we review a subset of these approaches.

Shock Graph Matching

The general aim in shock graph matching [136] is to match shock graphs extracted from binary shapes. The matching algorithm computes two similarity measures as follows.

The first similarity measure is a topological structure similarity which captures how much the derived graphs are similar to each other. Given two shock graphs, a bipartite graph is made between nodes of their DAGs. The eigenvalue-value sum of a sub-DAG of a given shock graph with computed eigenvalues of its corresponding submatrix is invariant to any similarity transformation applied to the submatrix. In a DAG representation, the TSV is defined as the vector of eigenvalue-sums derived from the corresponding adjacency matrix for the sub-DAG of the considered node [136]. Each edge is weighted based on the structural similarity between nodes; the weight is the normalized length of the difference of their topological signature vectors

$$w = \frac{|\mathbf{t}_1 - \mathbf{t}_2|}{\max(|\mathbf{t}_1|, |\mathbf{t}_2|)}$$

where $\mathbf{t}_1$ and $\mathbf{t}_2$ represent the corresponding TSV vectors of the two nodes of a considered edge. The best matching of a maximum weighted bipartite matching is when the sum

of the values of the edges is maximized. Shock graphs are represented using a $\{0, 1\}$ adjacency matrix, with 1's indicating adjacent nodes in the tree form of the DAG. The matching algorithm used is a greedy algorithm that has the benefit of finding the largest maximal matching in polynomial time. The similarity is computed by matching a query with a model node and then normalizing by the number of matched nodes according to the order of the model graph.

The second similarity measure captures the geometric between skeletal points on the nodes of each given DAG. To determine the similarity between nodes, Siddiqi et al. [136] uses nodes that represent curve segments of first-order and third-order shock points. Now, given a query and a model, the algorithm tries to fit the query to the model by allowing curve segments of the model to shrink or grow to include the query data points. The main assumption here is that the derivative of the radius function can be related to the object angle, which has a relationship to spoke vectors that connect the bi-tangents to their associated skeletal point.

Treating shock graphs as directed acyclic graphs (DAGs), a DAG matcher is needed to match a query shape with other shapes. Here, the DAG matcher receives two DAGs as input and computes a value representing their similarity, as well as a list of corresponding nodes in the two DAGs. This analysis considers both topological structure (Γ) and geometric information (Δ) associated with shock graphs' vertices. Each of these two measures return a value normalized in the interval $[0\ 1]$. The final similarity score is a

weighted combination of these two

$$S(G_1, G2) = \tau\Gamma(G_1, G_2) + (1 - \tau)\Delta(G_1, G_2),$$

where $S(G_1, G2)$ represent the similarity between DAGs derived from two given shapes, and τ is a tunning weight in the interval $[0\ 1]$. At the end of the process, a list of corresponding nodes and a similarity measure are provided.

Editing Shock Graphs

Sebastian et al. [127] proposed a novel edit distance-based approach to shape matching using shock graphs. Their method is based on an estimate of the distance between two shapes by the cost of the least action path deforming one to the other. According to their method, all shapes with the same shock graph topology belong to one equivalence class, called a *shape cell*.

Deformations are made by a sequence of shock graph transitions, thus, all deformations with the same sequence of shock graph transitions are equivalent. Here, shock graph transitions are edit operations on the shock graphs. In this approach, given two shapes with their corresponding shock graphs, a globally optimal transition sequence between their shock graphs is sought. The general approach follows a number of steps. First, a set of shapes is considered, where each shape is a point in shape space. Then the shape space is partitioned based on the shock graph topology. In this space, each deformation sequence between two shapes is a path between their corresponding points (see Figures 2.5 and 2.6a).

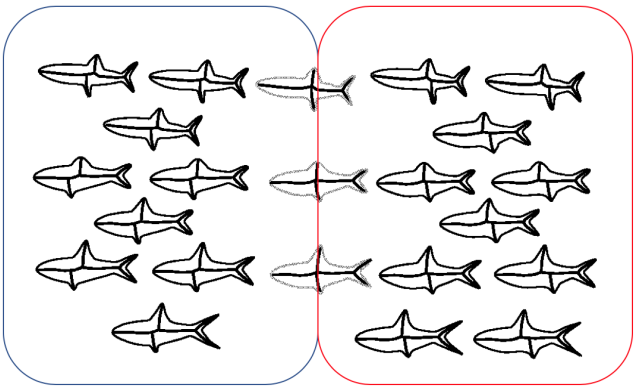
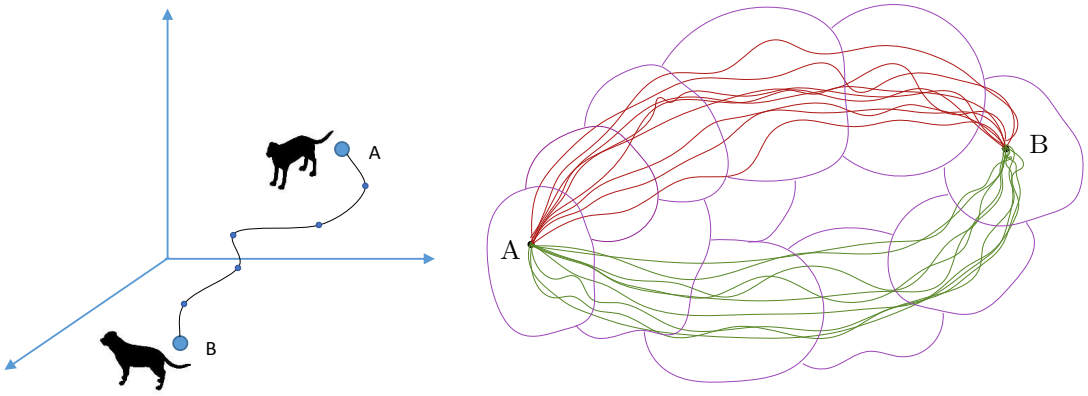


Figure 2.5: This figure illustrates two samples of shape cells, where shapes with same shock graph topology are grouped within a box. Some shapes would lie along the border of these cells, as shown above (Adapted from [127]).

The number of deformation paths between two shapes is infinite, and finding the optimal deformation path is intractable. In the second step of their approach, to make the problem tractable, they discretize deformation paths (see Figure 2.6b).

The third step is to apply a graph edit distance algorithm to find the optimal transition path among all the possibilities. The Levenshtein distance or edit-distance algorithm was proposed originally to compare character strings. To apply the edit distance approach to compare shock graphs, four groups of edit operations on shock graphs are used: a) Slice: where a shock branch is deleted and the adjoining two are merged; b) Contract: where a shock branch is deleted between degree-three nodes; c) Merge: where two branches are combined at a degree-two node; and d) Deform: an edge is replaced by another, having different attributes. By using a dynamic-programming based edit-distance algorithm, an optimal deformation sequence is found in polynomial-time.



(a) (b)

Figure 2.6: (a) There are an infinite number of deformation paths between shape A and shape B, where these deformations can be characterized by a sequence of transitions. (b) A set of deformation paths going through the same set of shape cells comprise a shape deformation bundle (Adapted from Sebastian *et al.* [127]).

Path Similarity Approaches

In path similarity approaches, introduced by [163, 5, 6, 162], the general idea is to match the shortest paths that connect skeletal endpoints. This is done by mapping the dissimilarity between two given skeletons to the dissimilarity measures computed between each of the two sets of nodes. For two given skeletal graphs G and G' , with respectively m and n nodes, a dissimilarity matrix $C(G, G')$ with $m \times n$ elements is constructed as follows.

First, a matrix $pd(v, v')$ is computed for every two matching nodes, (v, v'). Each element of $pd(v, v')$ is a dissimilarity value, e_{ij}, measuring the distance between two paths, one originating from v and ending at the node with index i from graph G and the other one originating from v' and ending at the node with index j from graph G'. The matrix $pd(v, v')$ is a dissimilarity matrix from which the best candidate paths for the matching

process can be found. The best matching path is a sequence of nodes matched along with the matrix $C(G, G')$ with a matching score that is considered as a similarity measure. In [5] this is accomplished using the optimal sequence bijection elastic matching algorithm, which is in the spirit of other dynamic programming approaches.

To obtain a similarity measure between two shapes, these methods consider the shortest path between the skeletal endpoints. These shortest paths are compared while considering that the structural graphs of different examples may vary. Although the shortest path descriptors can result in precise matching for many examples, the approach is more of a technically engineered system than a topologically justified method. It is unclear whether such methods are capable of handling transformations which can change the topology of the skeleton to a great extent. It is also not yet well established how these methods handle the matching of skeletal graphs having different numbers of nodes.

2.2.2 Non-Skeletal Graph-Based Shape Matching

Having reviewed selected skeletal graph-based methods for shape matching we now discuss a selection of shape matching methods used for the problem of object recognition and categorization that do not rely on skeletal graphs.

3D Object Categorization, Localization, and Pose Estimation

Savarese and Fei-Fei [124] proposed an approach for 3D class modeling of objects using appearance and 3D geometric shape information. In their approach, a given object is decomposed into large parts, where each part is a collection of scale-invariant feature

transform (SIFT) features [92]. Such features are qualitatively invariant to changes in scale, in-plane rotation, and lighting and can be used as key points in the matching of different views of the same object. Canonical views are found, and parts are related by their mutual appearances (see Figure 2.7). Similar to aspect graphs, this work also emphasizes representing stable views rather than parts.

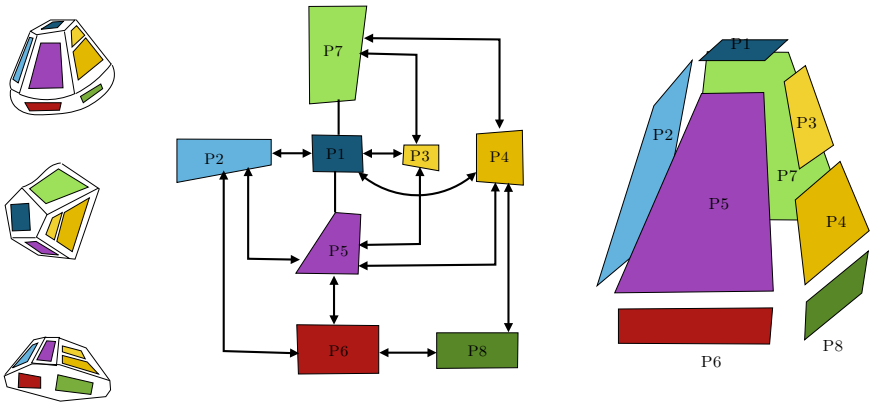


Figure 2.7: LEFT: a hypothetical object category. CENTER: model visualization I RIGHT: model visualization II. (Adapted from Savarese and Fei-Fei [124])

The decomposition process is applied to new images by recognizing parts and selecting a model that best accounts for their appearances. Then, for each pair of images of the same example, first a set of shared SIFT features is found (as M) and then the random sample consensus (RANSAC) algorithm is executed to find a group of pairs that transform together. Finally, close-together parts of M are grouped into candidate parts.

For a new query, the image recognition procedure follows four steps. First, SIFT features are extracted. Second, the 5 best canonical part matches are obtained by using scanning windows. Third, for all pairs of such parts, for each model, the algorithm scores how

well the model accounts for their relative appearances. Finally, the model with the highest score is selected.

Visual Learning and Recognition from Appearance

As discussed earlier, recognizing objects from images as well as estimating their 3D pose is a problem of long-standing interest in computer vision. The hope here is that by learning from distinct views one might be able to achieve recognition of nearby views efficiently. Such methods must be trained systematically on these distinct views and the typical strategy taken is to sample these views uniformly. Murase and Nayar [98] presented a method based on this intuition. Their approach tries to recognize a given object in an image and then estimate its pose in the 3D scene just by using appearance information.

Murase and Nayar [98] highlight four properties of the appearance of an object in an image: 1) shape 2) reflectance 3) pose 4) illumination. Their learning process is parametrized by pose and illumination since these vary with different views of the same object. This leads to a set of images of an object, obtained by varying pose and illumination. After obtaining such a set of images, and normalizing the scale and brightness, an eigenspace is constructed by computing the most prominent eigenvectors, and all images are then projected on to this subspace. The Euclidean distance in that eigenspace is counted as a measure of the similarity between images. Afterward, the approach uses cubic-spline interpolation to compute a manifold from these discrete points, and an object universal eigenspaces are made. The object eigenspace considers only images of a specific object and the universal eigenspace contains all images of all objects.

2.2 Object Recognition and Matching

The recognition phase involves a) segmenting the considered object b) normalizing the segmented image as before c) projecting that onto the universal eigenspace, where the closest manifold to the projected point identifies the object, and d) projecting that onto the associated object eigenspace, where the closest point on that manifold can be used to estimate the pose.

This has proved to be an effective method for appearance-based recognition from large databases in real-time. It also does not require significant low-level processing and does not compute geometric features. One of the drawbacks of this method though is that every time that a new object is added to the database, the entire eigenspace must be re-computed. Moreover, the approach requires that the objects to be recognized are seen in their entirety, without occlusion. In general, the method is suited to controlled imaging conditions.

Shape Context

A general theme in shape matching is that of finding correspondences between two shapes. To be able to find such correspondences one needs to have a matching framework that measures the similarity between candidate point sets extracted from those shapes. To this end, Belongie and Malik [8] proposed an approach based on a representation called the *Shape Context*. In this method, a shape context descriptor is a local feature attached to each point of the extracted contour of a given image. The shape context feature is defined as a histogram that counts how many contour points are present in the neighborhood of a considered central point.

Given a set of points $P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\}$ which are extracted from a given object's

boundary, the shape context of a point $\mathbf{p}_i$ is a coarse histogram h_i of the relative coordinates of the remaining $N-1$ points (see Figure 2.8),

$$h_i(k) = \#\{\mathbf{m} \neq \mathbf{p}_i : (\mathbf{m} - \mathbf{p}_i) \in \text{bin}(k) \text{ and } \mathbf{m} \in P\}.$$

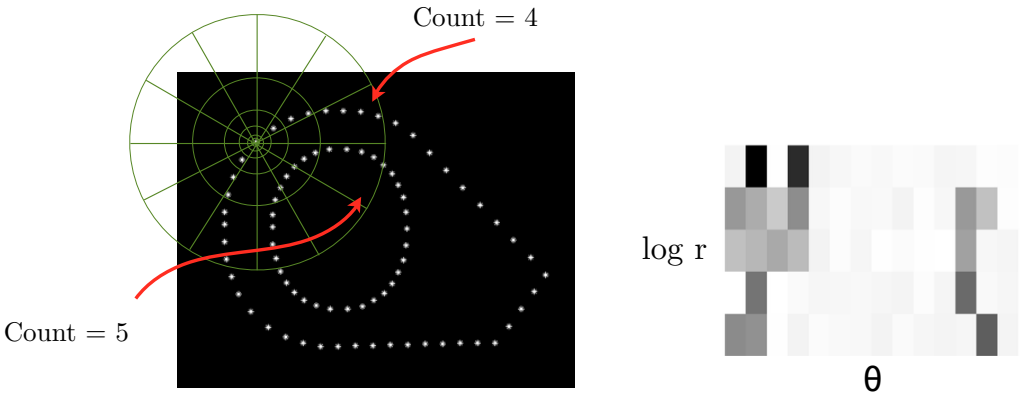


Figure 2.8: LEFT: Compact representation of the distribution of the points relative to a candidate point, where the number of points inside each bin is counted. RIGHT: An example diagram of log-polar histogram bins. (Adapted from Belongie and Malik [8]).

To be able to compare two points from two different shapes, the method suggests a matching cost. Let us consider point $\mathbf{p}_i$ from the first shape and point $\mathbf{m}_j$ from the second shape. The cost of matching between these two points, $C_{ij} = C(\mathbf{p}_i, \mathbf{m}_j)$, is defined as:

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)},$$

where $h_i(k)$ and $h_j(k)$ denote the K-bin normalized histogram at $\mathbf{p}_i$, and $\mathbf{m}_j$, respectively. By this definition, finding the correspondences turns into a least cost assignment problem,

where the goal is to minimize the total cost of matching

$$H(\pi) = \sum_i C(\mathbf{p}_i, \mathbf{m}_{\pi(i)}),$$

subject to the constraint that the matching be one-to-one, i.e., π is a permutation. By definition, the shape context is invariant to translation and scale, and by expressing point locations with respect to a local tangent frame it can be made invariant to rotation also.

Shape Classification using Inner Distance

Ling and Jacobs [86] extended the shape context idea [8] by replacing the original Euclidean distance in their work by a notion of inner distance. Considering some landmark points within a given shape, inner distance is defined as the length of the shortest path between these points. Inner distance is a shape descriptor that is robust to articulation and can capture part structure to some extent. The shape is represented the same way as with the shape context descriptor, with the difference that the bins are made using inner-distance rather than Euclidean distance. The only landmarks used in this method are boundary points. The tangential direction at the starting point of the shortest path between two points is assumed as the relative orientation between them and is called the inner-angle. Let us assume a 2D shape $\mathcal{O}$ as a connected and closed subset of $\mathbb{R}^2$. Given any two points, $(\mathbf{p}_1, \mathbf{p}_2)$, inside the considered shape the inner distance between these two points, $d(\mathbf{p}_1, \mathbf{p}_2, \mathcal{O})$, is defined as the length of the shortest path connecting $\mathbf{p}_1$ and $\mathbf{p}_2$. When $\mathcal{O}$ is convex, the inner distance will reduce to the length of the connecting line between the two considered points (the Euclidean distance).

Such a graph is constructed as follows. For each contour point, a graph node is considered. For each pair of such nodes, an edge is added between them if the connecting line between them lies entirely inside the shape $\mathcal{O}$, or in other words, their inner distance is equal to their Euclidean distance (see Figure 2.9). After that the connectivity graph is made, inner distances are computed by a shortest-path algorithm. Since the shape bound-

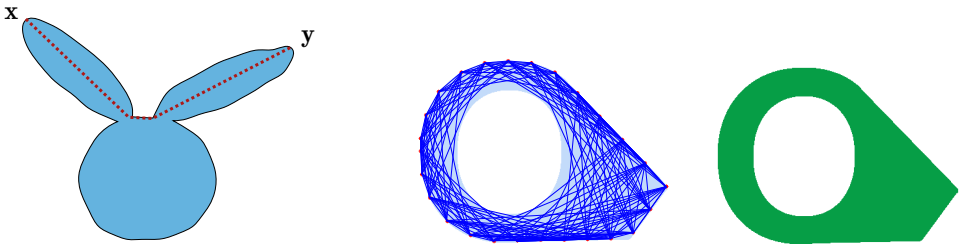


Figure 2.9: LEFT: An example of an inner-distance path between points $\mathbf{x}$ and $\mathbf{y}$s are considered from the boundary of the hole itself. In case there is more than one shortest path, one of them is selected. RIGHT: Holes are considered when computing inner distances, but no sample points are considered from the boundary of the hole itself. (Adapted from Ling and Jacobs [86]).

ary is sensitive to the noise,the contour is smoothed using a Gaussian weighted average of points in a small neighborhood. The rest of the matching, algorithm is similar to shape context matching and is carried out using bipartite graph matching.

2.3 Aspect Graphs and View-based Object Recognition

Recognizing a 3D object's class and its pose are challenging tasks in computer vision. Strategies to deal with the problem of object recognition fall into two categories. The first assumes a 3D object model as a whole and attempts to recognize the object from it. The other strategy is to consider 2D projections of a 3D object model from particular view-

points and then carry out recognition based on these projections. The second approach has received a lot of attention and many systems have been developed based on this strategy in the past decades.

The problem of understanding the pose of an object from a particular viewpoint dates back to Koenderink's notion of singularities of the visual mapping, where the term "aspect" was introduced [75]. An aspect refers to a region of viewpoints from which the same set of singularities are visible for a specific object. Assuming that each of these regions is represented as a node, these nodes form a graph where neighboring regions are connected by an edge, leading to an "aspect graph" [76]. The main goal of the aspect graph is to capture changes in appearance with viewpoint changes. Aspect graphs can be used to determine the "characteristic views" of an object, to provide a language for visual inspection (e.g. guiding a robot to a particular side of an object), to help sample views of objects for recognition and reconstruction, and for other view-based problems. Several approaches were developed in the late 80s and early 90s to compute aspect graphs [111, 58, 110, 23]. Eggert et al. [44] argue that the aspect graphs should be able to deal with different resolutions of a 3D object model.

While aspect graphs attracted a lot of attention in the community initially, they gradually lost their favor due to a variety of reasons, including the lack of available practical implementations. In a workshop panel report [48], several drawbacks of aspect graphs are discussed. These include the dependence on object part complexity, the overall size of the representation and finally the challenges of computing them from projections in the presence of noise.

2.3 Aspect Graphs and View-based Object Recognition

One way to address the problems with aspect graphs is to create viewpoint space partitions, where aspect regions are formed using a notion of similarity between views rather than using singularities of the visual mapping [34], as illustrated in Figure 2.10. In

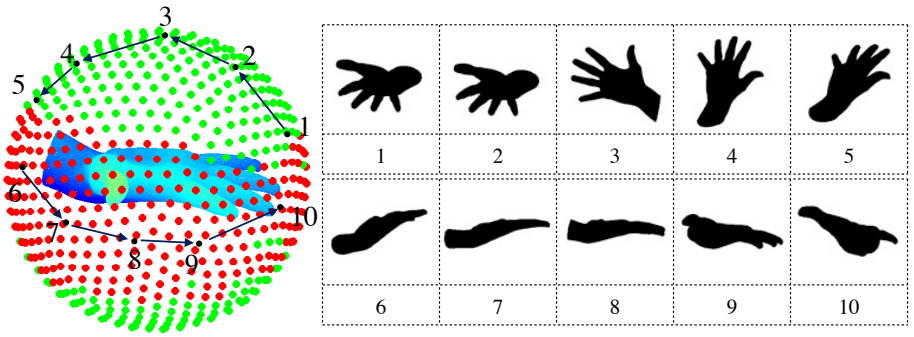


Figure 2.10: Assuming a view sphere around a 3D object model, as in the hand example here, viewpoints can be clustered into regions where similar views fall into the same partition.

[34] a new view-based technique to recognize 3D objects from 2D views is introduced, where each view on the view sphere is an aspect. Adjacent views are then merged if the projections are similar enough, using a similarity measure based on both an edit distance between shock graphs and a distance based on curve matching. To compare two regions, the algorithm specifically considers average pairwise similarities between their centroids and merges them if this value is below a certain threshold. This is an iterative process which stops when no two neighboring regions that can be merged exist anymore. At the matching time, given a query view, that view is matched against the centroid of each aspect region and those regions which are distant are discarded. The query is then matched exhaustively against all the views in each surviving region.

2.4 Environment Mapping and Topological Matching

Computer vision has made important contributions to robotics, where one of the major goals is to make machines that perceive their environments and can understand their surroundings. Computing a robot's pose and the map of the environment it is exploring at the same time, or simultaneous localization and mapping (SLAM), is one of the most fundamental problems in robotics. The two important components of SLAM are *localization*, which is the estimation of the robot's location and *mapping*, which is the task of generating a map by deploying suitable algorithms. During mapping, the robot must estimate the landmarks given its poses and for localization, the robot must estimate its poses given the landmarks. This makes SLAM a chicken and egg problem as a map is needed for localization and a pose estimate is needed for mapping. Several constraints can exist in the environment mapping problem that could make the problem quite challenging, such as not knowing the environment that a robot wants to explore in advance, or not having a distance orientation metric. These constraints add difficulties to achieving an executable solution unless one augments the robot with disambiguation capabilities, such as by using makers. Other than augmentation, exploration also provides important places and paths and from the exploration patterns, and one can create an abstract representation of the explored environment in the format of a topological representation. A topological map represents the environment as a graph of nodes and edges. Nodes in this graph represent places, landmarks or goals, and edges represent a path between two nodes that the robot can use for navigation. To implement control laws that are simple and have a correlation with human behavior, several roboticists have suggested topological representations for

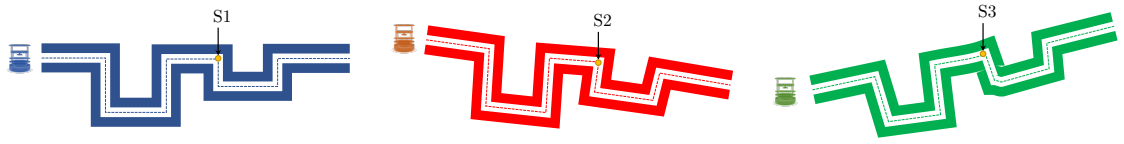
task [82, 25, 41].

In topological environment mapping, the key idea is to maintain a currently known sub-graph of the environment that a robot is exploring. The basic operation here is to select an unvisited node (or path) from a known graph at every time step and to then send the robot to explore the novel territory. This is a simplistic scenario as the hard part is to distinguish between visited and unvisited nodes. To solve this difficult problem, the system should be able to track the robot motion accurately according to its environmental setting to reduce the positioning error.

One common approach in topological mapping approaches is to guide the robot towards the mid-path between walls and/or obstacles. This ensures that the robot stays at maximally distant locations from sensed obstacles. Choset [31] proposed an incremental map construction algorithm that used the Generalized Voronoi Graph (GVG) representation to find the locus of points equidistant to two obstacles. This algorithm, when presented, was one of the first planning approaches that relied only on line-of-sight sensor information and offered completeness guarantees. At each time step, the robot generates a part of the roadmap edge and then follows that edge to explore the next segment. In this procedure, the robot traces an edge until it gets to a new branching node that brings new edges to explore for the mapping algorithm. As the motivation for this work is based upon sensor-based planning for robots, the algorithm considers a set of distance functions where each is between the robot and one particular single obstacle. This set is then formalized in the form of the distance between a point that represents the robot position and the set of all obstacles in the environment. Based upon this formalization, points that make the path for

the robot to explore are characterized and obtained in the form of either a two-equidistant edge point or a three-equidistant branch point. The use of a GVG in this approach makes it possible to have a concise representation of the workspace or configuration space. The algorithm presented in this work uses just sensor data to compute the GVG in the process of map-making.

Positioning error is one the intrinsic challenges in the task of SLAM, where the assumption is that the robot does not have an external positioning device, nor the luxury of engineered landmarks placed in its free space. This is where localization ideas could come to help and rescue the algorithm. One of the strategies for localization is to find points with a similar topology, which is the process of bringing sensed-data into alignment. To gain further intuition about this problem, consider an unknown environment with three different possible scan outcomes from three different scan poses, as illustrated in Figure 2.11. In the ideal world, we want these three scans to be exactly the same as the ground truth map, but this is not the case here. Clearly, when the sensor-based data is not aligned, we want to have an algorithm that can find the corresponding matching point from one scan to another, i. e., when matching the maps from each of these scans, S1, S2, and S3 map to each other.



(a) Scan from pose 1 (b) Scan from pose 2 (c) Scan pose 3

Figure 2.11: Three scans of the same unkown environment, each obtained from a different starting pose.

Choset and Nagatani [30] included metric information in the GVG topological representation of the environment, to be able to localize the robot on a partially constructed map. Again, as the topological representation makes it possible, a simple control law guides the robot to unseen areas, and the robot gradually explores the environment and completes the topological map. The main contribution here is the use of a GVG representation in the form of a graph structure that makes it possible to do graph matching for the task of localization. One of the other localization tasks in topological mapping is to disambiguate places that appear indistinguishable. To achieve this goal, Werner et al. [153] suggested using neighborhood information extracted from the sequence of observations along with the use of a particle filtering technique to detect loop-closures. Using neighborhood information is particularly helpful if the robot visits a physically different place which appears to be the same as a previously explored environment.

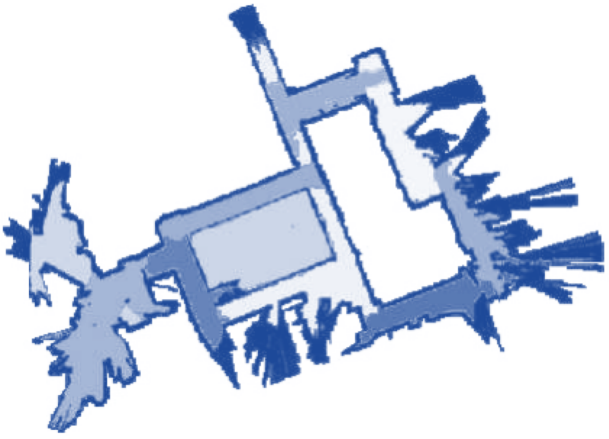


Figure 2.12: An example environment broken into submaps by spectral clustering (adapted from [20]). Each shade shows a different submap obtained by the algorithm.

Brunskill et al. [20] proposed a spectral clustering-based method to break a sensed

grid map into sub-map segments and then train a classifier to recognize graph submaps from laser signatures. The algorithm uses a spectral clustering algorithm, which is a graph partitioning algorithm [164], to subdivide the environment that robot is exploring into a set of submaps of points that are spatially correlated. In this method, each grid map is considered a graph node in the graph that is going to be clustered and edges are placed between all pairs of mutually visible locations. The clustering algorithm is applied at each time step of the map creation. As the robot explores more unvisited areas the map grows and the whole process is repeated (see Figure 2.12). To detect when the robot is re-visiting a submap, the process learns to label each visited submap by using the Adaboost algorithm, whereby when there are two submaps with associated classifiers that overlap sufficiently, the algorithm merges those submaps.

As reviewed earlier, medial axes are prominent geometric structures in computer vision and robotics since they relate to local axes of mirror symmetry. There are several methods that use Voronoi diagrams for skeleton computation [103, 129, 130] due to the theoretical relationship between them [47]. In particular, the vertices of the Voronoi diagram of a set of boundary points converge to the exact skeleton as the sampling rate increases under some appropriate smoothness conditions [126]. As a result, one can use the medial axis in mapping and planning problems. Xu et al. [161] proposed a motion planning algorithm of a polygonal object through a set of planar obstacles. Their algorithm involved a two-disk motion planning strategy to guide the robot within its free space between the obstacles from a starting point to an endpoint. Masehian et al. [97] offered an online motion planning algorithm whereby a medial axis of the workspace of the robot is incrementally constructed as the robot explores the unknown environment. This the-

sis also presents an online environment mapping algorithm based on AOF skeletons that shows robustness in both real environments and simulated ones. We also provide a point matching approach for topology mapping and finding loop closures that are based on the spectral point correspondences algorithm of Lombaert et al. [90]. These developments are discussed in Chapter 4.

2.5 Contour Geometry in Scene Categorization

Vision feels natural and effortless: light hits our eyes, and we appear to understand our real-world environment almost instantly. Yet, the neural computations underlying visual perception are not fully understood. As an example, scene categorization is still a challenging task for vision systems. Due to its numerous applications in photo search and surveillance, scene classification has become one of the hot topics in computer vision in recent years, and several visual descriptors have been developed to solve the problem of scene classification in computer vision in the past decades. The first step in most of these algorithms is that of extracting visual features from some input training images. Those features can then be fed to a classifier to categorize new input data and produce semantic class label predictions. This makes the role of feature extraction crucial for scene categorization tasks. Researchers have developed a range of computational algorithms to replicate the gist recognition ability of human vision for image classification. Great progress has been made in identifying the basic visual features that are extracted from the visual input, such as oriented edges [67], corners [87], spatial frequency content [121, 105], and disparity [14]. At the other end of the processing pipeline, brain areas have been identified that

are dedicated to the processing of objects [95], faces [70], or places [46]. The processes involved in the intermediate-level grouping of visual features is less well understood.

Humans have the ability to classify a photograph of a natural scene after a brief exposure [113, 142, 146]. In spite of a sizable literature studying this phenomenon [104, 144, 36, 155], there is no consensus on what accounts for this ability. Even if we replace photographs of natural scenes with line drawings of the same content, experiments show that observers can rapidly recognize the class label of line drawing [12, 150]. This provides strong evidence that the information stored in the contours that make outlines of the regions of a natural scene is quite rich and can be used for scene categorization. Furthermore, Berman et al. [9] demonstrates that scene content is carried primarily in the high spatial frequencies. In fact, the high-pass images used in the latter study closely resemble line drawings. Walther et al. [151] find that the neural patterns in photographs and line drawings are very similar, which results in similar underlying category-specific representations. As line drawing contours of natural scenes appear to have a lot of information compared to scene photographs, we choose to review algorithms for scene recognition that are layout based and use contour-based representations in their proposed approaches, but not color or texture.

One of the very first studies on the informational aspects of visual perception goes back to Attneave [3]. This study on the most important Gestalt perceptual principles concludes that information along visual outlines of shapes is not necessarily distributed uniformly, but rather, the regions with higher magnitude curvature contain more information than the rest. Feldman and Singh [50] bring this suggestion into a formal derivation by

presenting an expression for information as a function of contour curvature. By considering the signed curvature function along closed contours of object boundaries, Feldman and Singh [50] drive an equation that relates the information magnitude and the signed curvature along with the contour points. According to this equation, the information quantity is minimized when the curvature is zero and it is greater along negatively curved (concave) portions of the contour than along positively curved (convex) portions (see Figure 2.13).

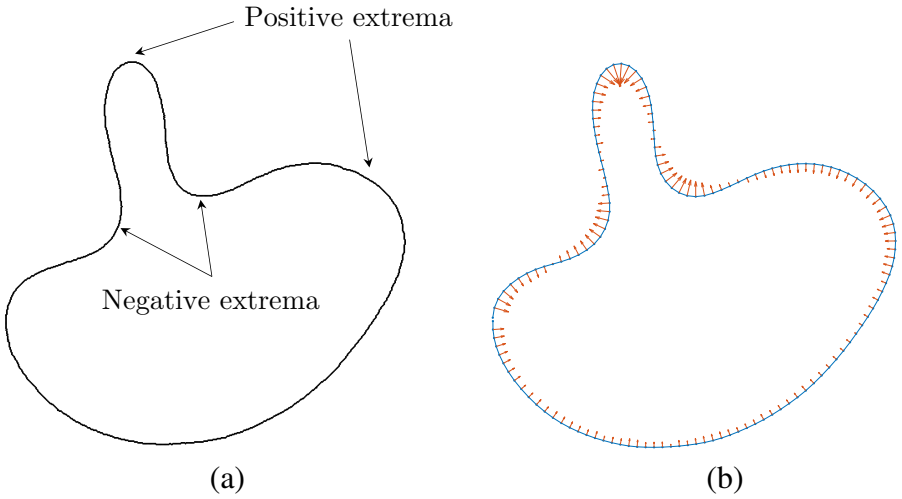


Figure 2.13: (a): A shape with curvature extrema marked, including both positive (convex) extrema and negative (concave) extrema (i.e., minima of signed curvature). (b): The same shape with contour information (surprisal) plotted (adapted from Feldman and Singh [50]).

The role of negative curvature extrema has also been prominent in discussions of proposals for the part decomposition of silhouettes. For examples, Hoffman and Richard's use a transversality argument to associate negative curvature extrema with part boundaries [64]. Siddiqi et al. derive grouping principles for such extrema based on how they interact through the interior of an object via the notion of part-lines and smooth contin-

uation on one side [132, 134]. Finally, Singh and Hoffman further examine perceptual aspects of parts related to interactions between negative curvature extrema, based on a related notion of part cuts [65, 138].

Inspired to further study perception of line drawings of natural scenes, Walther and Shen [150] looked at structural characteristics of contours (orientation, length, curvature) and contour intersection (types and angles) specifically those which were computationally obtainable, and could contain data about the class of scenes. In a six-alternative forced-choice scene categorization experiment by human observers, they found that spatially related measures that describe non-accidental junctions and curvature in line drawings of real scenes have a higher impact on outcomes of visual image categorization to human behavior than properties like direction or contour length.

Principles of grouping [77] for the perceptual organization have inspired many studies in human vision, psychophysics, and computer vision to examine which visual cues contribute to the perception of objects and scenes. In addition to the structure properties listed in [150], other visual perception characteristics such as symmetry could be investigated to analyze their role in perception. Symmetry can be viewed in two ways. The first is when all the points involved in the description of an object or scene are considered, and a global notion of symmetry is explored. The second is where a subset of the image that locally contributes to a set of symmetric elements, is considered. Often, by this subset, we shall refer to a continuous section of a shape's contour on either side of a symmetry axis. Local symmetry is widely used for shape description, as it provides a compact coding and a useful tool for recognition and categorization.

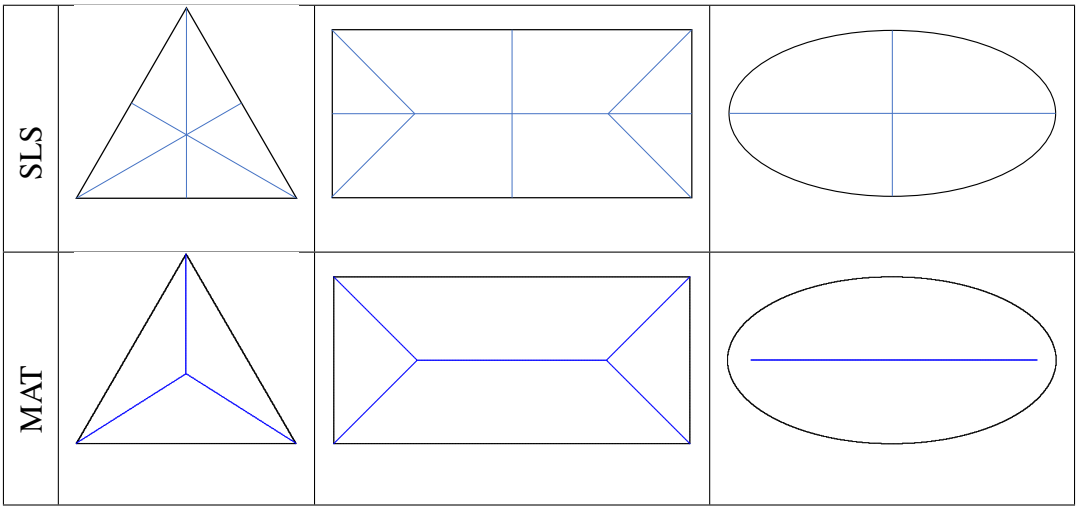


Figure 2.14: Comparing the Smoothed Local Symmetry (SLS) method of Brady and Asada [19] and the Medial Axis Transform (MAT) (see Section 2.1) for three common shapes.

Brady and Asada [19] introduce Smoothed Local Symmetries (SLS) as a representation of two-dimensional shape that is both contour and region-based and can provide a description of objects in terms of parts and sub-parts and their links to each other. The computation of SLS is carried out in two stages. First, local symmetry is determined by finding locally symmetric points on the shape border (two points are locally symmetric if both angles of the segment that connects them and the normals to the boundary at those points are the same). Second, a notion of a skeleton is formed by considering the union of all symmetry axes, where an axis corresponds to the formation of maximal smoothed loci of local symmetries. In Figure 2.14, we compare the SLS approach with the Medial Axis Transform (MAT). One problem with the smoothed local symmetry method is that it may end up creating redundant spines for the shapes which do not exhibit perfect local symmetry. Furthermore, computing SLS can be expensive since one must test all pairs of

border points.

As both symmetry and curvature extrema provide geometric descriptors for shape, Leyton [83] focused on the duality between these two entities. He provided a duality theorem, which reads as follows:

> If we say that curvature extrema are of two opposite types, either maxima or minima, then the theorem states: Any segment of a smooth planar curve, bounded by two consecutive curvature extrema of the same type, has a unique symmetry axis, and the axis terminates at the curvature extremum of the opposite type [83, page 327].

This theorem glues the information driven from extrema of curvature [3] to that of symmetry, one of the crucial Gestalt-based organizing principles of shape [154] (see Figure 2.15).

Later, in [84], Leyton develops a formal grammar to infer the relationship between any two smooth shapes. This process-based grammar uses symmetry axes (MAT and SLS) along with curvature extrema [83] to develop inference rules that are then put together to make a process-diagram. Specifically, the grammar includes six operations to generate all possible process extrapolations between two shapes in which the process-diagram of one shape could be transformed into the other one. The process grammar for shape provides a psychologically meaningful expression for the relationship between shapes.

Although Blum's MAT provides an appealing representation for visual form problems, this tool suffers from the fact that a tiny protrusion on the boundary can dramati-

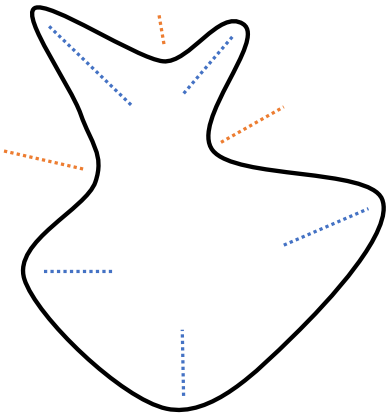


Figure 2.15: An illustration of Leyton's Symmetry-Curvature Duality Theorem [83]. Every pair of consecutive curvature extrema of the same type on the boundary leads to a symmetry axis between. The interior symmetries are shown in blue and the exterior ones in orange.

cally change the MAT's structure. Current methods to compute the medial axis can deal with small boundary perturbations, but these methods are typically used in conjunction with salience measures for medial points, in applications. Examples of popular salience measures include the object angle [102], the AOF itself [38], the boundary to axis ratio [17] and more recently, work by Katz and Pizer on a notion of visual conductance [71]. In particular, Katz and Pizer propose a perceptual part-decomposition approach which is based on calculating a measure of the amount of "substance" information at each medial point.

Part II

Flux Graphs, View Sphere Partitioning and Environment Mapping

3

Flux Graphs for View Sphere Partitioning

The contents of this chapter are largely based on the article "View sphere partitioning via flux graphs boosts recognition from sparse views" [118]. This work was carried out prior to the current wave of deep learning approaches to object recognition, and its specific goal was to explore silhouette based 3D object recognition from 2D views. With the advances that have been made in the production of vision sensors in recent years, the recognition of real 3D objects seems more possible than ever. Using the algorithm presented in this chapter, one could apply these ideas to recognize 3D real-world objects from their outlines. With the rich information contained in the silhouette of a 3D object, enterprise systems that process large datasets can take advantage of view-based recognition strategies to handle sparse views.

View-based 3D object recognition requires a selection of model object views against which to match a query view. Ideally, for this to be computationally efficient, such a selection should be sparse. To address this problem, we partition the view sphere into regions within which the silhouette of a model object is qualitatively unchanged. This is accomplished using a flux-based skeletal representation and skeletal matching to compute the pairwise similarity between two views. Associating each view with a node of a view sphere graph, with the similarity between a pair of views as an edge weight, a clustering algorithm is used to partition the view sphere. Our experiments on exemplar level recognition using 19 models from the Toronto Database and category-level recognition using 150 models from the McGill Shape Benchmark demonstrate that in a scenario of recognition from sparse views, sampling model views from such partitions consistently boosts recognition performance when compared against queries sampled randomly or uniformly from the view sphere. We demonstrate the improvement in recognition accuracy for a variety of popular 2D shape similarity approaches: shock graph matching, flux graph matching, shape context-based matching, and inner distance-based matching.

3.1 Introduction

View-based object recognition has seen many recent advances with the current state of the art systems achieving promising category-level recognition results on large databases of real images. An effective strategy here is to learn suitable models from images taken in a controlled fashion of several exemplars from a particular category [52, 124, 160]. The hope here is that by learning from distinct views, one might be able to achieve recognition

of nearby views efficiently. Such methods must be trained systematically on these distinct views and the typical strategy taken is to sample these views uniformly, as in the original principal component-based method of Murase and Nayar [99]. As our community attempts to grapple with the full complexity of object recognition from arbitrary views, we face the challenge that such methods may not easily generalize to views that have not been seen before, at least not without a prohibitive amount of training.

To gain further intuition about this problem consider the sample silhouettes of a dog seen along two different trajectories in Figure 3.1. In the first trajectory, we move from a top rear view to a top front view (views 1-5) and in the second we rotate around the dog showing side views (views 6-10). Qualitatively views 6 through 10 are those from which the dog is most easily recognizable in that prominent parts (the head, limbs, and tail) remain visible. On the other hand views in the first trajectory are more challenging to recognize because parts are foreshortened or occluded. This example points to the need for judicious view sphere sampling to achieve recognition from sparse views. Approaches which train on views sampled uniformly on the view sphere do not reflect the complex relationship between view stability and surface area on the view sphere.

The problem of defining regions on the view sphere with qualitatively similar views of a 3D object has a long history in the computer vision community, dating back to Koenderink's notion of transitions on the view sphere as one moves from one location to another, as signaled by the appearance or disappearance of singularities of the visual mapping [76]. This lead to a tremendous interest in the computer vision community in computing *aspect graphs* designed to capture changes in appearance with viewpoint changes,

3.1 Introduction

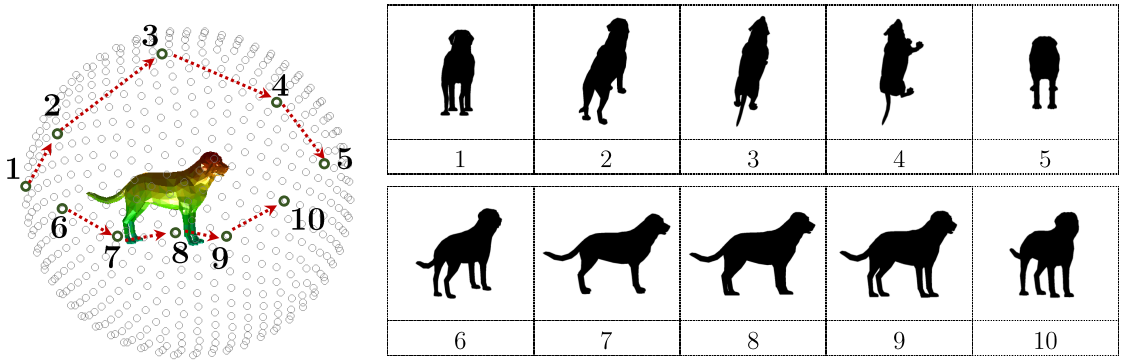


Figure 3.1: Silhouettes of a dog are shown for viewpoints along two trajectories on the view sphere: (1,2,3,4,5) and (6,7,8,9,10).

as reviewed in Chapter 2. A great deal of conceptual progress was made in the late '80s and early '90s with techniques developed for computing aspect graphs of polyhedra [23], of curved surfaces described algebraically [108] along with considerations for the role of scale [44]. Whereas aspect graphs had intuitive appeal, they lost favor for a variety of reasons [48]. These included the fragility of the concept itself for facilitating object recognition (e.g. not all singularities of the visual mapping are visually salient) and the difficulty of computing it for general 3D object classes. The object recognition community in computer vision has since shifted to appearance-based representations for category-level recognition from natural images of objects, see for example [112] and [37]. Current approaches typically combine robust feature detection, the modeling of local geometric relations between derived "parts" and advanced statistical machine learning methods for classification [91, 49, 52, 53, 160]. The underlying representations are viewpoint dependent and thus careful training is required for them to handle arbitrary views of 3D objects. As these methods advance, from both computational efficiency and storage efficiency considerations, judicious sampling of the view sphere resurges as a problem of interest.

3.1 Introduction

One way to approach this problem is to focus on the appearance of 2D silhouette shape as the viewpoint changes. Promising steps were taken in this direction by Cyr and Kimia [34]. In their approach, the views on the view sphere are treated as nodes of a graph, with the similarity between two views providing an edge weight. Their similarity measure uses both an edit distance between shock graphs and a distance based on curve matching. Each node on the view sphere is initially a cluster (region), and clusters get merged when they are geographical neighbors and the average pairwise similarities between their centroids are below a certain threshold. This process is iterated until it converges. Once the regions have been obtained, at matching time, a query view is matched against the centroid (characteristic view) of each region for a particular model. Those regions whose centroids are distant from the query are discarded. The model is then matched exhaustively against all the views in each surviving region.

Motivated by the success of medial representations for computing part-based representations of silhouettes for matching [127, 5, 133] and the promise of silhouette-based similarity for 3D recognition in [34], we consider the problem of recognition from sparse views using skeletal graphs. The specific question we look at is that of generating view sphere partitions from which to efficiently (and not necessarily exhaustively) sample model views when faced with a new query. As such our goal is complementary to that of [34], but our methods are different. Specifically, we employ an average outward flux-based skeleton [38] together with a novel measure for simplification which leads to a directed flux graph. In our work, we employ a hierarchical clustering algorithm to obtain a view sphere partition where views in each partition are similar. Unlike the region-growing approach of [34], our partitions are based on a spectral decomposition strategy, which leads

to partitions whose nodes are not necessarily geographical neighbors. More importantly, the context of our experiments is different and somewhat complementary to that of [34]. Specifically, we evaluate the benefit of selecting model views from the partitions when only a sparse number of model view queries are allowed, whereas, in [34], the matching experiments use all the model views in each surviving cluster. Our main contribution is to show that hierarchical view sphere partitioning boosts 3D object recognition performance in the scenario of matching against a sparse number of model views. Our experiments also demonstrate the importance of selecting centroids of the clusters during matching time for the four shape matching algorithms we have evaluated: shape context-based matching, inner distance-based matching, flux graph matching, and shock graph matching.

This chapter is organized as follows. We discuss flux graphs for 2D shape representation in Section 3.2. We then develop the view sphere partitioning strategy in Section 3.3, where a clustering algorithm is employed on a graph whose nodes are views and whose edge weights are pairwise similarities between flux graphs. This leads to partitions within which the model silhouettes are qualitatively similar. We demonstrate the utility of these partitions for selecting model views in Section 3.4 by comparing this to the alternatives of random or uniform sampling. We conclude with a discussion in Section 3.5.

3.2 Flux Graphs

We adopt the AOF approach of [38] to compute the flux-based skeleton (see Section 2.1). We then consider the degree to which the area reconstructed by the maximal disk associated with a skeletal point is unique. Specifically, for each skeletal point, we compute the

fraction of area of its maximal disk that does not overlap with the disk of a skeletal point on any other branch. This relative area contribution measure is novel to the literature and is particularly simple to compute while being effective. The measure decreases monotonically as one approaches a branch point, as illustrated by the three sample calculations in Figure 3.2 (left). Numerous other salience measures for medial loci have been proposed [133] but most combine a notion of boundary-to-axis ratio, which can be delicate to compute, with the object angle, and require choices of thresholds and parameters to be tuned. The monotonic decrease in the relative area measure as one approaches a branch point suggests a more robust simplification procedure, which is to move in the direction away from a branch point and retain only those skeletal points with relative area measure above a threshold. This process is illustrated for the dog shape in Figure 3.2 (right). The threshold can be chosen adaptively to ensure that the desired percentage of the original shape's area is captured, as illustrated in Figure 3.3 (left). For all the experiments and examples in this chapter, we require at least 95% area coverage. The parts reconstructed by each black skeletal segment are shown in distinct colors, with skeletal segments which have been removed by the simplification process shown in light pink.

Any skeletal branch includes two ends, each of which can either be a branch point or an end point. If we assume our boundary is a C^1 curv, we can assum that the spoke end points associated with maximal inscribed disks are different for each skeletal point. Using this assumption, we can prove that the uniqueness of area measure will not have a strict local maximum along a medial axis branch. First of all, the uniqueness of maximal inscribed disk measure for all branch points is always zero, and this is due to the fact the maximal inscribed disk at a branch point is shared by different branches. Now to prove this

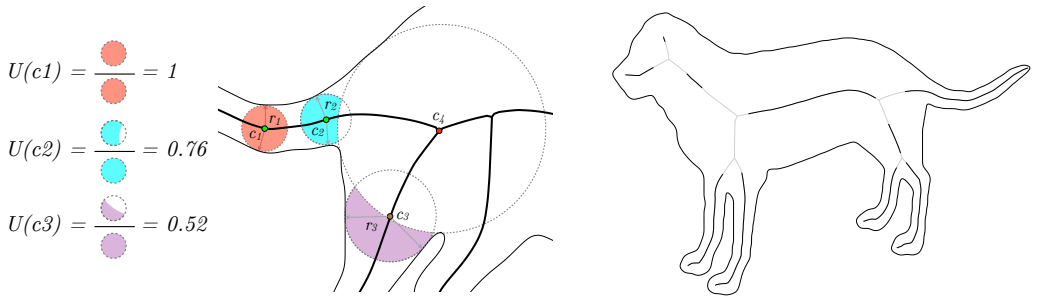


Figure 3.2: LEFT: For three skeletal points, $c1, c2, c3$, we calculate the fractional area of the maximal inscribed disk that does not overlap with that of a skeletal point on any other branch. This relative area measure decreases monotonically as one approaches a branch point. RIGHT: The skeletal points with fractional maximal disk area above a threshold are shown in black, with other flux-based skeletal points shown in grey. The threshold is chosen so that the black skeletal segments reconstruct at least 95% of the shape's area (see Figure 3.3 (left)).

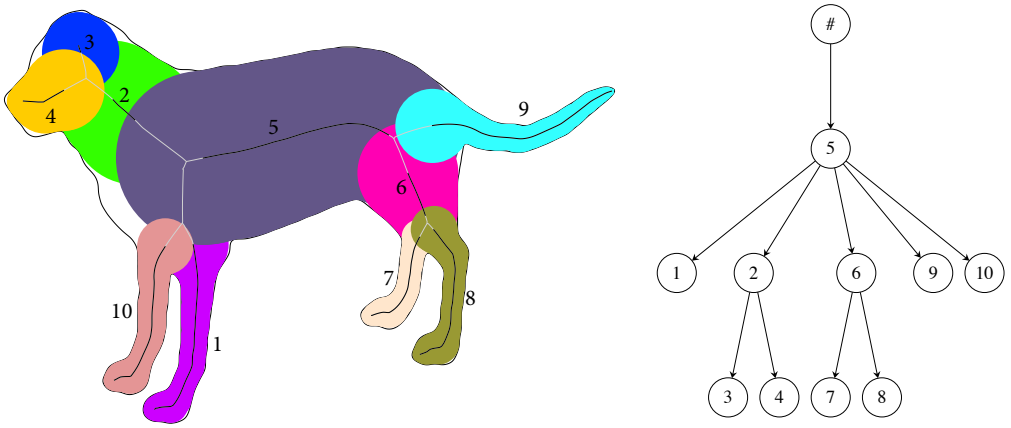


Figure 3.3: LEFT: The nodes corresponding to the retained skeletal segments (black) are shown in different colors, each representing a union of medial disks. RIGHT: The corresponding flux graph. The dummy node ♯ carries no geometrical information but serves as a parent to all the top level nodes.

monotonicity property, we will prove that as one follows a medial axis branch towards a branch point, the uniqueness of maximal inscribed disk measure can either monotonically

decrease/increase or or reach a local maximum, but never a local minimum. To prove this, we just consider two neighboring medial axis points where one of them is closer to a considered branch point on that medial branch. Without loss of generality, let us consider two neighboring medial axis points, $\mathbf{A}$ and $\mathbf{B}$ with their corresponding maximal inscribed disks that touch the boundary on unique spoke points $(\mathbf{S}^1_\mathbf{A}, \mathbf{S}^2_\mathbf{A})$ and $(\mathbf{S}^1_\mathbf{B}, \mathbf{S}^2_\mathbf{B})$ respectively. We also assume that by walking through the medial branch, we can get to the point $\mathbf{B}$ sooner than the point $\mathbf{A}$ from branch point $\mathbf{C}$ (as shown in Figure 3.4).

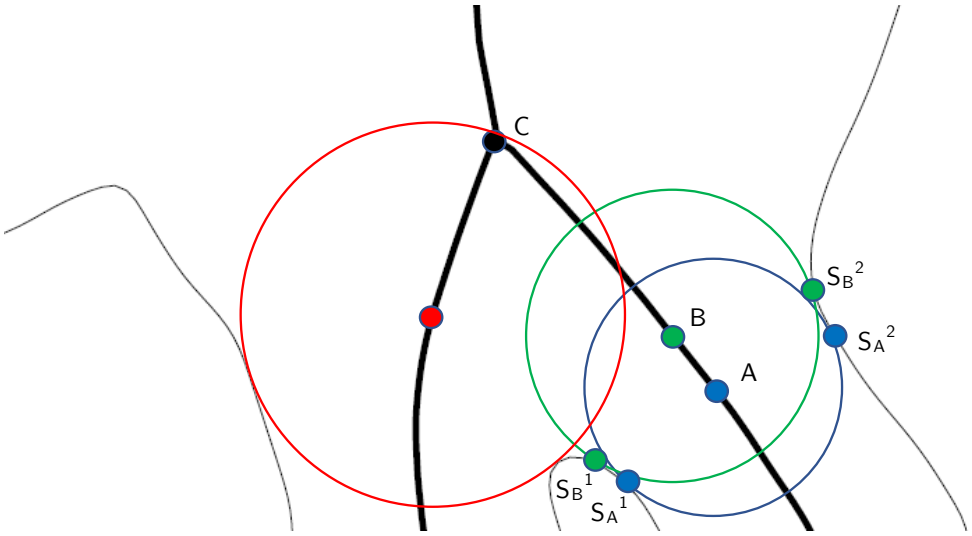


Figure 3.4: An example of how a disk from another branch can intersect with neighboring disks from a considered branch.

Now, we can observe that any maximal disk from other branches that connects to $\mathbf{C}$ will intersect more with the maximal inscribed disk at $\mathbf{B}$ (relative to the area of $\mathbf{B}$) than with the maximal inscribed disk at $\mathbf{A}$ (relative to the area of $\mathbf{A}$). In fact, if a disk from another branch intersects more with the disk at $\mathbf{A}$ than the disk at $\mathbf{B}$, it should intersect with either of the lines $\mathbf{S}^1_\mathbf{A}$ to $\mathbf{S}^1_\mathbf{B}$ or $\mathbf{S}^2_\mathbf{A}$ to $\mathbf{S}^2_\mathbf{B}$ which is not possible due to the definition

of the medial axis (that disk would no longer be a maximal inscribed disk).

The monotonicity property ensures that for each original skeletal branch at most one skeletal segment is retained. We can, therefore, associate each retained skeletal segment with the node of a graph. Then, using the topology of the original skeleton, for any set of adjacent nodes edges are placed in the direction from the node having the largest average maximal inscribed disk radius to the others. The one with larger magnitude is chosen as the parent and the other as the child. The resulting directed acyclic graph represents a hierarchy of parts, as illustrated for the dog shape in Figure 3.3 (right), which we dub the Flux Graph. As it turns out, this simplification procedure, based on relative area alone, is more robust than the strategy first proposed in [117].

3.2.1 Qualitative Stability with Viewpoint Changes

We provide a qualitative demonstration that flux graphs remain stable under small changes in viewpoint while providing an intuitive part structure. We consider the views in the second trajectory in Figure 3.1, and represent their respective flux graphs in Figure 3.5. For each view, the top row depicts the parts represented by each node of the flux graph, the second row the flux graph and the bottom row the shock graph. Changes to the flux graph typically occur when new parts, such as the leg, come into view (or disappear) but the overall graph structure is much simpler than that of the shock graph. This is expected because the shock graph utilizes and hence represents the entire skeleton, without any simplification.

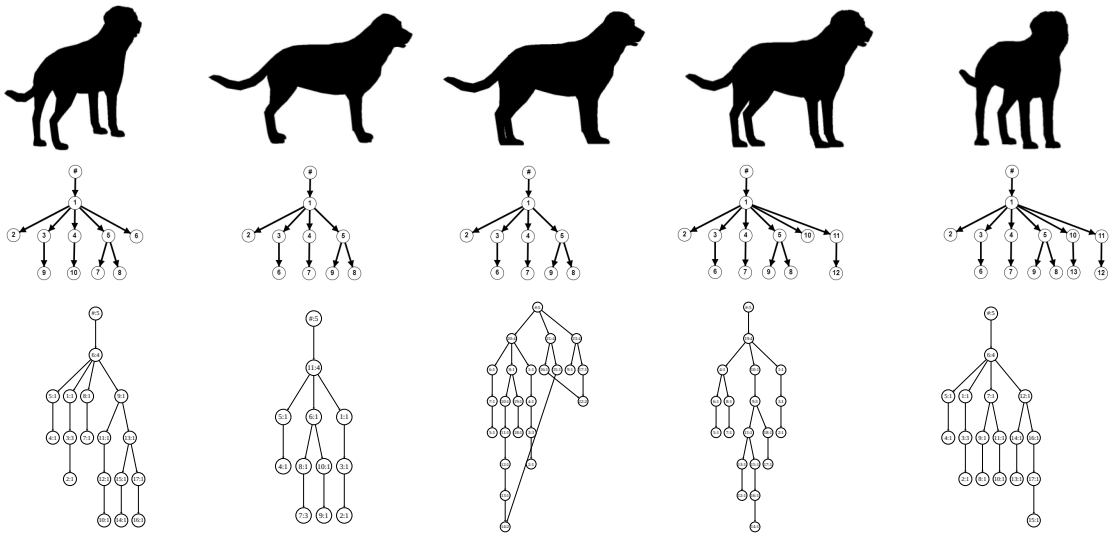


Figure 3.5: Top row: Side views of a dog, along a trajectory on a view sphere surrounding it. Middle row: The flux graph corresponding to the view in the top row. Bottom row: The shock graph corresponding to the view in the top row.

3.2.2 Flux Graph Matching

In the present chapter, we use the established method for matching directed acyclic graphs (DAGs) in [136] for flux graph matching. Given two flux graphs, a bipartite graph is hierarchically constructed between their nodes. Each edge of the bipartite graph is weighted based on the structural similarity between the nodes. This weight is based on the normalized difference between the topological signature vectors (TSVs) introduced in [136]. A maximum weighted bipartite matching is then carried out such that the sum of the values of the edges is maximized. In a DAG representation, the TSV is defined as the vector of eigenvalue-sums derived from the corresponding adjacency matrix for the sub-DAG of the considered node. The matching algorithm used is a greedy algorithm by Macrini et al.

[94], which has the benefit of finding the largest maximal matching in polynomial time. The similarity is computed by matching a query with a model node and then normalizing that by the number of matched nodes according to the cardinality of the model graph.

The above structural similarity measure (Γ) is combined with a notion of the geometric similarity (Δ) between the parts corresponding to two nodes, where for the latter we use the elastic matching approach in [94]. Here line segments are fit through the skeletal points of a given node and then, for a given query and a given model, the algorithm tries to fit the query to the model by allowing line segments of the model to shrink or grow to include the query data points. The data points themselves encapsulate both positions along a skeletal branch and the radius of the maximal inscribed disk. The main assumption here is that the pattern of velocities and acceleration is invariant to small changes in viewpoint, where velocity and acceleration are defined as first and second derivatives of the radius along the medial axis.

Putting these measures of similarity together, a DAG matcher receives two DAGs G_1 and G_2 as input and computes a value $S(G_1, G_2)$ representing the similarity between them, as well as a list of corresponding nodes. Both Γ and Δ are in the interval $[0\ 1]$ and $S(G_1, G_2)$ is given by a weighted combination: $\omega\Gamma(G_1, G_2) + (1 - \omega)\Delta(G_1, G_2)$. Here ω is a tuning weight in the interval $[0\ 1]$.

As presented here, flux graphs are computable for all 2D bounded shapes and they lead to a hierarchy of parts. Moreover, the representation allows for common transformations and is stable to some boundary purturbations. Finally, being able to compare flux graphs, both with geometric similarity and structural similarity measures, provides us a notion

of scale in a coarse to fine sense. Therefore, flux graphs are excellent candidates for the problem we are addressing in this chapter, i.e., they satisfy a number of the desirable criteria of a good representation, as discussed in Chapter 1.

3.3 View Sphere Partitioning

We use the flux graph to represent the silhouette seen from each view on the view sphere of an object. We then create a dense view sphere graph by associating each view with a node and placing an edge between each pair of nodes. To each edge, we associate a weight based on the similarity between the views using the DAG matcher described above. The problem of view sphere clustering can now be treated as a clustering problem on the view sphere graph. Our goal is to find clusters of view sphere points with high within-cluster similarity. Such clusters should, in principle, correspond to regions of the view sphere within which the silhouette shapes are similar. To this end, we employ a clustering algorithm but in a hierarchical fashion. Intuitively, the idea is to recursively partition clusters until a particular derived cluster has a high enough within-cluster similarity and is then treated as a leaf node of a view sphere graph. The final set of clusters then correspond to the set of leaf nodes.

Within cluster similarity is, of course, maximized when the clusters are very small, so we impose a minimum cluster size for partitioning. Given a view sphere Γ, we find a suitable number of clusters for decomposition (Algorithm 2) and then the hierarchical clustering algorithm is applied recursively (Algorithm 1). A stopping condition (Algorithm 3) is imposed whereby a cluster is no longer divided if it is small (in practice, its

Algorithm 1 Hierarchical view sphere clustering

Declaration of variables
Γ: View Sphere
$C_i = \{p_{i1}, ..., p_{ik}\}$: a cluster which includes a set of points on the view sphere
procedure PERFORM_HIERARCHICAL_CLUSTERING(C_i,n)
    if SHOULD_CLUSTERING_BE_STOPPED(C_i) == false **then**
        $n \leftarrow$FIND_NUMBER_OF_CLUSTERS(C_i)
        $\{C'_1, ..., C'_n\} =$`CLUSTERINGALGORITHM` (Γ, n)
        for $1 := 1$ `to` n **do**
            PERFORM_HIERARCHICAL_CLUSTERING(C'_i,n)
        end for
    else
        return C_i as a leaf
    end if
end procedure
▷ Note that `clusteringAlgorithm` is a choice here. In this work, as mentioned in the text, we chose the Normalized Cuts algorithm.

Algorithm 2 Find the number of clusters for decomposition

procedure FIND_NUMBER_OF_CLUSTERS(C_i)
    M $\leftarrow$ maximum number of branches per level
    $\mu \leftarrow$ threshold on average similarity for branching
    for $t := 2$ `to` M **do**
        $\{C'_1, ..., C'_t\} =$`CLUSTERINGALGORITHM` (C_i, t)
        $m \leftarrow$ weighted average of the average pairwise similarities between all nodes in the C'_js.
        if $m > \mu$ **then**
            return t
        end if
    end for
    return 0
end procedure
▷ Note that each cluster is weighted by the number of nodes on that cluster, divided by the total number of nodes on the view sphere.

size is already $< 10\%$ of the view sphere) or if the average of the pairwise similarities within that cluster is above a threshold.

Algorithm 3 Determine whether a cluster is a leaf node

```
1:  procedure SHOULD_CLUSTERING_BE_STOPPED(C_i)
2:      τ ← a threshold on the cluster size
3:      ρ ← a threshold on the average similarities
4:      m ← average similarity between all pairs of nodes in C_i
5:      if size(C_i) < τ then
6:          return true
7:      else if m > ρ then
8:          return true
9:      end if
10:     return false
11: end procedure
```

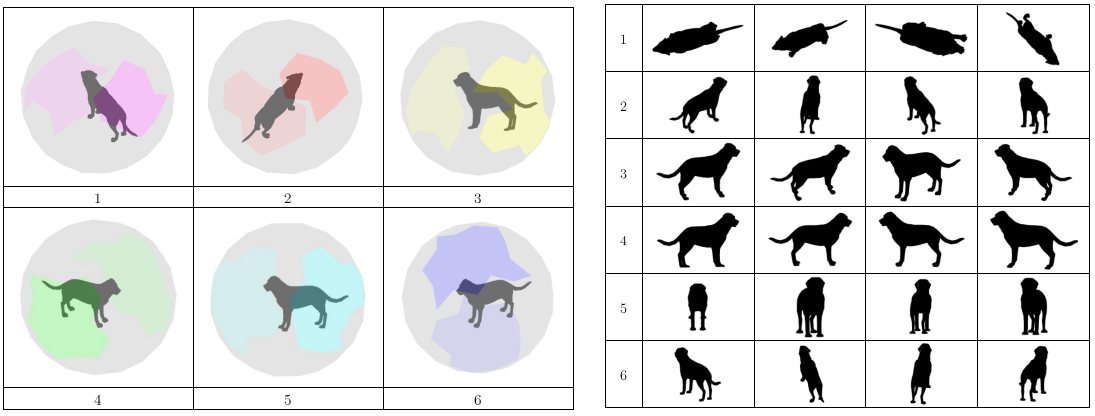


Figure 3.6: LEFT: Views of the dog on the view sphere belonging to the same cluster are shown as colored regions with distinct colors in panels $(1-6)$. RIGHT: Silhouettes are shown for views sampled from each of the clusters on the left. See text for a discussion.

Using 128 equispaced viewing directions, Figure 3.6 (left) illustrates the view sphere clusters obtained for the dog using the method above. Regions of the view sphere belonging to the same cluster are shown in the same color, in separate panels $(1-6)$. For this

example, there is an inherent symmetry to the clusters, such that each is composed of two diametrically opposed regions on the view sphere. To give a sense of how the views within a particular cluster are similar with regard to part structure and part shape, Figure 3.6 (right) depicts sample silhouettes for trajectories taken within each cluster. Cluster 1 contains views from above or below such that the body is elongated and the tail is visible but the limbs are occluded. In contrast, clusters 3 and 4 depict side views in which the limbs are visible and are extended.

3.4 Experiments

We evaluate the efficacy of using our view sphere partitions for recognition from sparse views. In our recognition experiments, we consider a query view and compare matching it with model views chosen from the view sphere clusters proportional to their size, versus matching it against model views chosen randomly from a uniform distribution of views on the view sphere. We have also conducted experiments where the model views were chosen to spread across the view sphere evenly, by using a particle repulsion method such that the distance between all pairs of neighboring closest model views was approximately the same. However, we found that in all cases, random sampling from a uniform distribution outperforms this particle repulsion strategy. This is likely because spreading the views across the view sphere leads to a higher probability of selecting model views that are ambiguous in that they are less representative of a particular object. We compare four different silhouette matching approaches: flux graph matching, shock graph matching, shape context-based matching, and inner distance-based matching. The skele-

tal graph-based matching experiments are carried out using Macrini's publicly available directed acyclic graph (DAG) matching package: http://www.cs.toronto.edu/~dmac/download.html, which contains an implementation of the approach in [136]. For shape context-based matching [8], we use the original implementation by Belongie et al., available at https://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc_digits.html, and for inner distance-based matching [86], we used Ling and Jacobs' implementation, available at http://www.dabi.temple.edu/~hbling/code_data.htm.

Both shape context-based matching and inner distance-based matching rely on finding correspondences between sample points from the boundaries of the two silhouettes. In the former, the shape context descriptor is based on a histogram of the contour points present in a local polar neighborhood of each sample point, considering both Euclidean distance and relative position. Two silhouettes are then matched using the Hungarian algorithm for finding the lowest cost matching between the two sets of histograms. This approach is extended in [86] by replacing the Euclidean distance by a notion of inner distance. The inner distance is defined as the length of the shortest path within the silhouette between two boundary points, and it provides some robustness to part articulation. A silhouette is represented in the same way as when using the shape context descriptor but with the difference that the bins in the histogram are constructed using inner distance in place of Euclidean distance. The tangential direction at the starting point of the shortest path between two points is treated as the relative orientation between them and is called the inner angle. Two silhouettes are matched by finding the lowest cost matching between the two histograms, but allowing for boundary sample points to be skipped but with an

associated penalty.

3.4.1 Recognition Performance

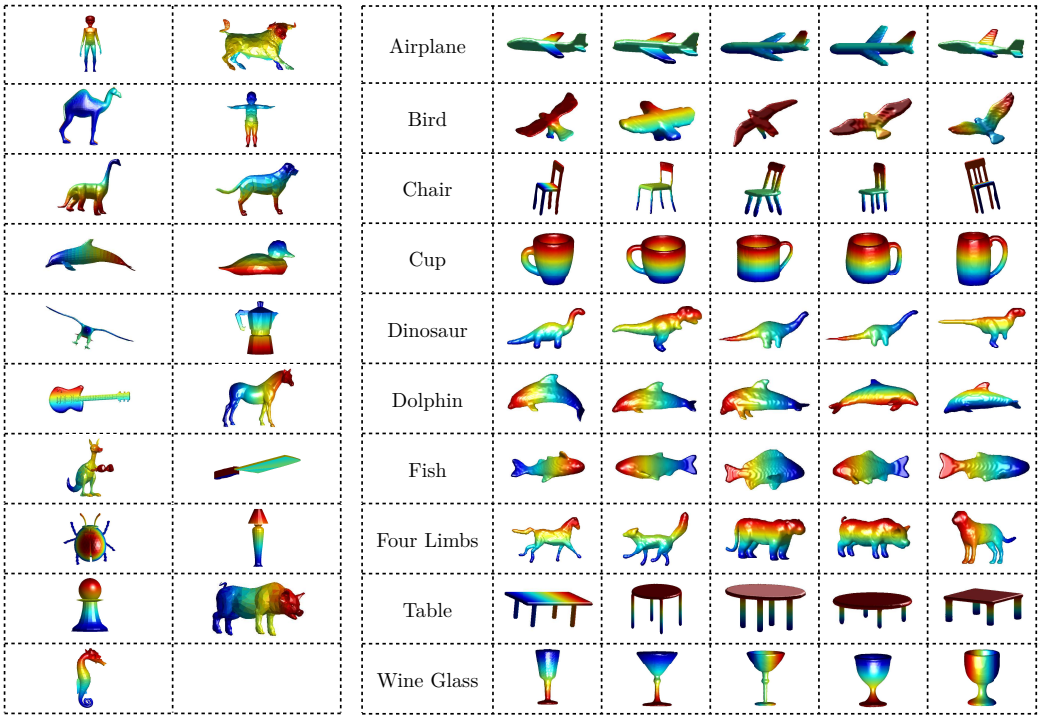


Figure 3.7: LEFT: The 19 object models in the Toronto database which we use for our experiments in exemplar level recognition. RIGHT: A selection of the 150 models from the McGill 3D Shape Benchmark which we use for our experiments in category level recognition. In total we have 10 object classes with approximately 15 models in each.

Exemplar Level Recognition

We begin with the exemplar level Toronto database of 19 models (Figure 3.7 left) because of the available published experimental results on skeletal graph-based recognition in [94]

as views are successively removed for these objects. In that work, Macrini *et al.* remove up to 75% of the views on the view sphere for a subset of 13 object models from this database, with 128 views of each. They report a graceful degradation in performance with occlusion, achieving 87% correct recognition with the shock graph as the representation. We carry out similar experiments evaluating flux graph matching, shock graph matching, shape context based matching, and inner distance-based matching. Our goal is to demonstrate that in a sparse model view selection scenario, sampling these views from our view sphere partitions can boost recognition performance in *all* cases.

The 19 Toronto database models are scaled to fit in a view sphere having radius 1. On this sphere, we sample 128 views using a particle repulsion based approach for node placement [122]. For each object, we construct a view sphere tree with the following choices of parameters. First, the minimum cluster size for partitioning is chosen to be 12 viewpoints. Thus, any cluster smaller than this size is not further partitioned. Second, the average within-cluster pairwise similarity threshold, where the pairwise similarities are obtained using the DAG matcher, is set to 0.65 (on a scale from 0 to 1). Thus any cluster with an average pairwise similarity higher than this threshold is not further subdivided. These view sphere partitions are computed offline. In our experiments, we have evaluated both the Normalized Cuts clustering algorithm of Shi and Malik [131] using the package at `http://www.cis.upenn.edu/~jshi/software/` and the K-medoids algorithm based on the implementation at `http://www.mathworks.com/help/stats/kmedoids.html` for clustering. Both lead to very similar recognition performance for all four shape matching methods; so both here and for category-level recognition we present results based on the Normalized Cuts algorithm.

3.4 Experiments

At recognition time, we match a query view against a sparse set of 8 sample views for each model, representing about a 94% reduction in view sphere nodes. These samples are chosen proportional to cluster size, i.e., the number of samples taken from each cluster depends on the area of that cluster relative to the entire view sphere. The first sample from a cluster is chosen to be its centroid, while the rest are picked randomly. The centroid of a cluster is defined as that view whose average pairwise similarity with all other views in the cluster is maximum. In what follows, we refer to this geographical area-weighted view selection strategy as partition sampling.

In the first set of experiments, we match all 19×128 views from the database to the 19×8 sample views, taking care not to ever match a silhouette with itself. We find the best match and define it to be correct if it corresponds to a view of the same object. We then repeat this experiment and define the best match as the one with the majority vote among n closest samples, with n varying from 1 to 30. In the second set of experiments, we match all 19×128 views from the database to 8 views of each of the 19 objects that are now chosen at random. The results of random sampling (dashed lines) versus partition sampling (solid lines) are reported in Figure 3.8 for shape context matching (green), inner distance matching (purple), flux graph matching (blue), and shock graph matching (red). It is striking that partition sampling consistently boosts recognition performance over random sampling, i.e., the solid lines are above the dashed counterparts. There is also strong evidence that in a scenario of recognition from sparse model views, skeletal (flux graph or shock graph) matching out performances shape context based matching and inner distance-based matching. The results for shock graph matching are slightly better than those for flux graph matching because the elastic matching approach to the geometric

similarity between nodes in Macrini's DAG matcher is optimized for shock graphs.

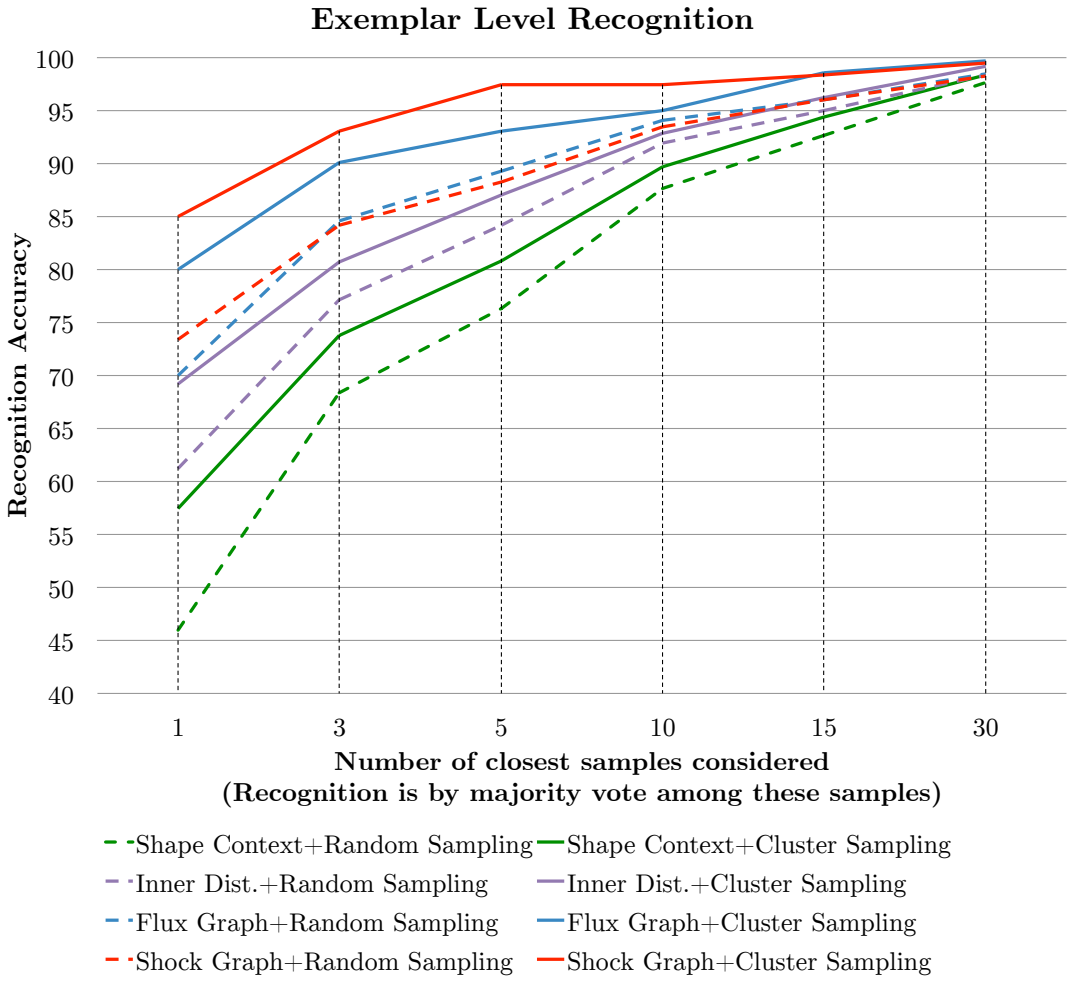


Figure 3.8: We compare view sphere partition sampling (solid lines) of model views against random sampling (dashed lines) of model views for four shape matching methods applied to an exemplar level recognition task. See text for a discussion.

Category-Level Recognition

We now carry out a more ambitious set of recognition experiments at the category level using a selection of 150 models from the McGill 3D Shape Benchmark. The process is

exactly the same as that used for exemplar level recognition above, except that we now have 150×128 query views to match. A recognition trial is assumed to be correct when the best match is with a model within the same category. This task is inherently more challenging than the exemplar level recognition task, due to the similarity in shape between some of the object categories (e.g. four-limbed and dinosaur, as well as fish and dolphin) and also due to the significant variation in shape within an object class (e.g. four-limbed). The results of random sampling (dashed lines) versus partition sampling (solid lines) are reported in Figure 3.9 for shape context matching (green), inner distance matching (purple), flux graph matching (blue), and shock graph matching (red). Once again, it is striking that partition sampling boosts recognition performance over random sampling. Averaging over all the 150×128 queries the improvement is from 59.60% to 71.70%, using shape context matching, from 67.04% to 77.52% using inner distance matching, from 85.52% to 89.18%, using flux graph matching and from 86.56% to 91.20%, using shock graph matching. The relative performance of the four matching methods is similar to that for exemplar level recognition, except that flux graph matching is now almost on par with shock graph matching.

Exploring Different Sampling Strategies

We now consider alternate model view sampling strategies for both the exemplar and category level recognition experiments, under the same sparse view scenario as before. We compare four different approaches. In the first (VS1), the model views are selected randomly from a uniform distribution on the view sphere. In the second (VS2), the model views are selected randomly from our view sphere partitions, but with the number of views

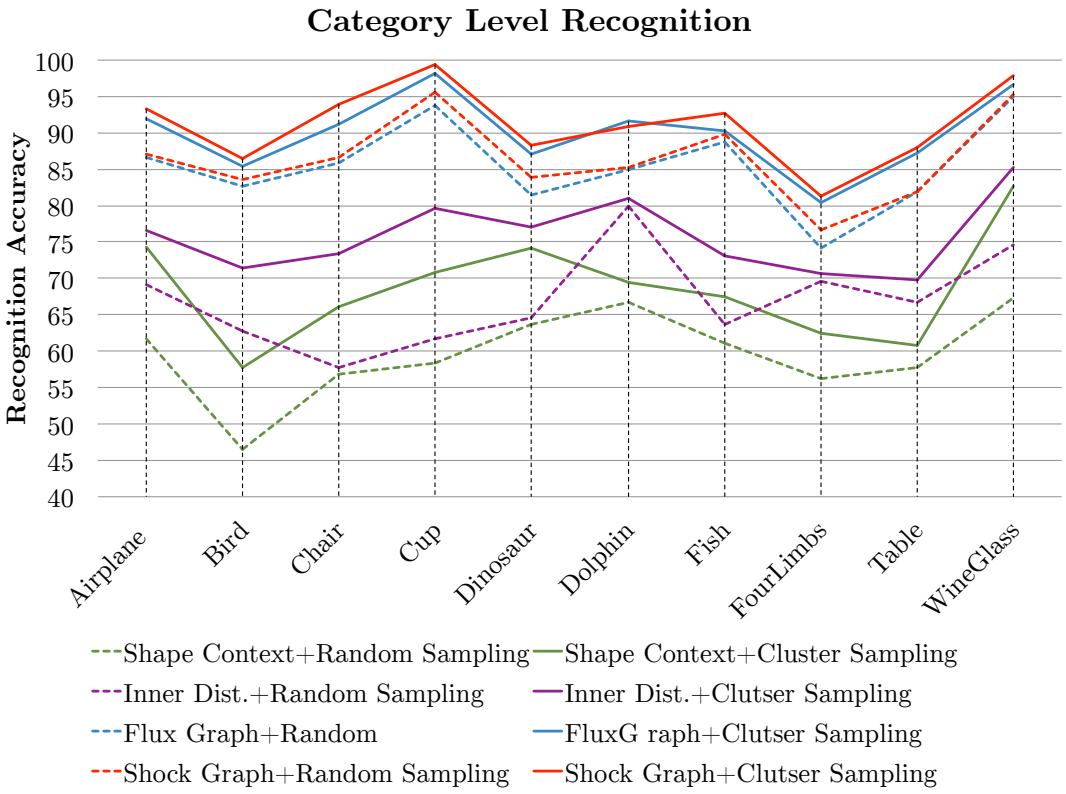


Figure 3.9: We compare view sphere partition sampling of model views against random sampling of model views for four shape matching methods applied to a category level recognition task. See text for a discussion.

from each partition being proportional to its size. The third approach (VS3) is similar to the second, with the exception that the first view from a partition is chosen to be its centroid. The fourth approach (VS4) is similar to the third except that now additional views from a partition, when required, are now selected in decreasing order of average pairwise similarity to the other views in that partition. As such the view sphere sampling strategies VS1 and VS3 are those used earlier in this section, but VS2 and VS4 are new. The results, averaged over all the objects, are presented in Figure 3.10 for the four shape

matching strategies at the exemplar level (left) and the category level (right). These results demonstrate the importance of choosing the centroid as the first sample view. Notably, as we move from random sampling (VS1) to random sampling from partitions but with the number of views proportional to partition size (VS2) the improvement is slight. However, when moving to VS3, where the first view from a partition is its centroid, we see a boost in performance of up to 10% in some cases. The additional benefit of selecting subsequent views from a partition in decreasing order of average pairwise similarity to the other views in that partition (VS4) is only slight. This is likely because in our sparse model view scenario not many partitions end up having more than one model view.

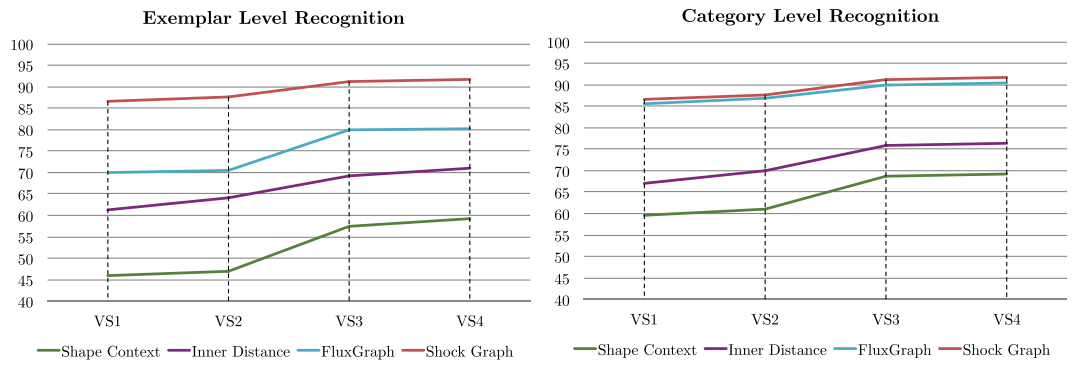


Figure 3.10: We compare recognition performance under different sampling strategies, with the results averaged over all the objects in the database at the exemplar level (left) and the category level (right). See the text for a discussion.

3.4.2 Flux Graphs versus Shock Graphs

Whereas shock graphs outperform flux graphs for exemplar level recognition, in part because of the optimization of the matcher for them, the narrowing of this gap for category-level recognition combined with the simplicity of flux graphs offers certain advantages.

Using the 19 models in the Toronto database, but now with 1000 views of each, and using a subset of 110 objects of the 150 used for category-level recognition from the McGill Shape Benchmark, but now with 1000 views of each, we compare shock graphs using the publicly available code at `http://www.cs.toronto.edu/~dmac/download.html` with flux graphs using a number of complexity measures: the average number of nodes, the average number of edges, the average depth, the average number of skeletal points, and the average TSV (topological signature vector) component values. The results presented as $\frac{\text{fluxgraph}}{\text{shockgraph}}$ ratios in Figure 3.11, show similar trends for both databases: flux graphs have 40% or fewer edges and nodes than shock graphs and 30% fewer skeletal points.

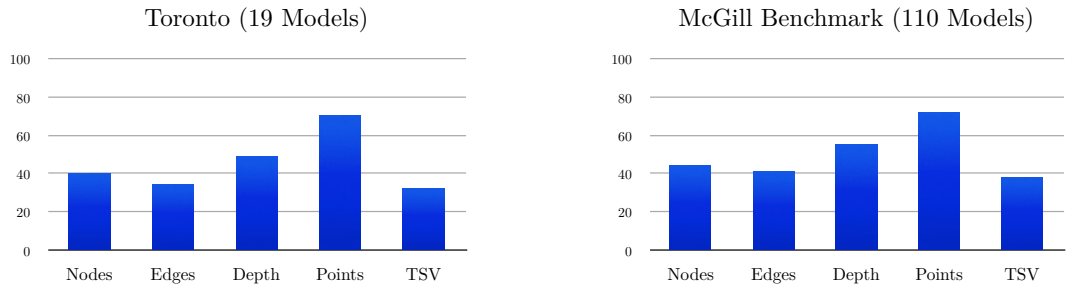


Figure 3.11: We plot the ratio of several complexity measures between flux graphs and shock graphs, as percentages, for the Toronto database of 19 models with 1000 views of each (19000 silhouettes in total) and for 110 models from the McGill 3D Shape Benchmark with 1000 views of each (110000 silhouettes in total). See the text for a discussion.

3.4.3 Running Time Complexity

We now consider how the matching methods evaluated in this chapter stand against one another in terms of running time complexity. The task of recognition is divided into two parts: representation creation and matching a query representation against another one.

For both flux graphs and shock graphs, a skeletal graph is constructed for each view and is stored in memory. For shape context and inner distance, we extended the original implementations to avoid repetition during the matching phase. In particular, for each silhouette, we compute the representation histograms first and store these in memory. Since each query silhouette is matched against a very large database of views, this preliminary step of precomputing and storing the histograms speeds up the matching phase considerably.

To measure the running time of these algorithms, we considered the following set up: 3 different views were selected randomly from each of the 150 models from the McGill 3D Shape Benchmark. One of these views was added to the query set and the other two to the model set. Thus, we had a total of 450 silhouettes. For each of the four methods, we then measured the time to (a) create the representations and (b) match the query against the model views. During the matching phase, we matched the 150 queries views to each of the 300 model views. The total running time, in seconds, is listed in Figure 3.12. All the experiments were performed on a PC with a 2.4-GHz Intel(R) Core(TM) i7-4700MQ CPU, 8 GB RAM, an NVIDIA GeForce GT 750M graphics card, and a 250-GB SSD disk. For these measurements, we used Windows 8.1 Pro. To generate a fair comparison, the input silhouettes were normalized to have the same resolution and an initial number of boundary points.

These results show that flux graphs and shock graphs are slower to compute than representations based on shape context or inner distance. However, flux graph and shock graph matching, using Macrini's DAG matcher, are significantly faster than inner distance-based matching or shape context based matching. This may in part due to the actual complexity

	Representation Creation	Matching
Shape Context	250.91s	12915s
Inner Distance	887.30s	61650s
Flux Graph	4241.25s	134.58s
Shock Graph	2942.67s	277.40s

Figure 3.12: Running time complexity for the four shape matching algorithms. See text for a discussion.

of the associated algorithms in terms of the number of operations but also in part due to the fact that the DAG matcher package is implemented in C/C++, while the others are designed to be user-friendly and use non-optimized code, e.g., based on Matlab.

3.5 Discussion

In the present chapter, we have demonstrated the promise of view sphere partitioning for 3D recognition from sparse views, by the hierarchical application of a clustering algorithm on pairwise similarities computed between flux graphs. Our experiments at the exemplar level on the Toronto model database (19×128 silhouettes) and at the category level on a selection of objects from the McGill 3D Shape Benchmark (150×128 silhouettes) demonstrate the consistent improvement possible by partition sampling of model views during the recognition phase, using the generated view sphere partitions. This improvement applies to each of the four shape matching algorithms we have evaluated: shape context based matching, inner-distance based matching, flux graph matching, and shock graph matching.

Our work suggests a number of fruitful directions for further research, having to

do with the use of precomputed view sphere partitions in online recognition scenarios. Clearly, there is rich information contained in the silhouette of an object by which to facilitate 3D recognition and with 3D point cloud data of real 3D objects now within reach of computer vision researchers via Kinect type sensors, offline computation of view sphere partitions is becoming feasible. We conjecture that as the object recognition community seeks to advance view-based recognition strategies to handle arbitrary but sparse views of objects, we will see a return to the application of the many good ideas that were alive two decades ago in the aspect graph literature.

4

Average Outward Flux Skeletons for Environment Mapping

The contents of this chapter are largely based on the article "Robust Environment Mapping Using Flux Skeletons" [116] which grew out of a collaboration with colleagues in the robotics groups at McGill University and the University of South Carolina. This work has been extended to include new experimental results on more complex environments, as well as a method based on spectral signatures for topological environment matching.

We consider how to directly extract a road map (also known as a topological representation) of an initially-unknown 2-dimensional environment via an on-line procedure which robustly computes a retraction of its boundaries. While such approaches are well

known for their theoretical elegance, computing such representations in practice is complicated when the data is sparse and noisy (see Section 2.4).

In this chapter, we first present the online construction of a topological map and the implementation of a control law for guiding the robot to the nearest unexplored area, first presented in [116]. The proposed method operates by allowing the robot to localize itself on a partially constructed map, calculate a path to unexplored parts of the environment (frontiers), compute a robust terminating condition when the robot has fully explored the environment, and achieve loop closure detection. The proposed algorithm results in smooth safe paths for the robot's navigation needs. The presented approach is an any-time-algorithm which has the advantage that it allows for the active creation of topological maps from laser-scan data, as it is being acquired. The resulting map is stable under variations to noise and the initial conditions. We also propose a navigation strategy based on a heuristic where the robot is directed towards nodes in the topological map that open to empty space. The method is evaluated on both synthetic data and in the context of active exploration using a Turtlebot 2. Our results demonstrate a complete mapping of different environments with smooth topological abstraction without spurious edges.

We then extend the work in [116] by presenting a topology matching algorithm which leverages the strengths of a particular spectral correspondence method, FOCUSR [90], to match the mapped environments generated from our topology making algorithm. Here, we concentrated on implementing a system that could be used to match the topologies of the mapped environment by using AOF Skeletons. In topology matching between two given maps and their AOF skeletons, we first finding correspondences between points on

the AOF skeletons of two different environments. We then align the (2D) points of the environments themselves. We also compute a distance measure between two given environments, based on their extracted AOF skeletons and their topology, as the sum of the matching errors between corresponding points. We evaluate our topology matching algorithm and demonstrate promising results on a few environments of increasing complexity, with simulated sensor noise.

4.1 Introduction

Our approach is based on AOF skeletons from two dimensional, dense, laser data. This fundamentally one-dimensional structure (embedded in 2D) constitutes a robust, efficient elegant representation that can be used for a range of navigation and localization tasks.

Topological representations have been proposed and employed in robotics for over 25 years [81, 25, 41] because of the potentially simple ensuing control laws, their relevance to human cognitive mapping, and their simplicity. At the core of many approaches to extracting topological representations from real environments is the calculation of points that are locally maximally distant from, sensed obstacles, yielding the medial axis. Such topological structures result in safe areas where robots can navigate without collisions. Indoor environments with long corridors, like those found in malls and office buildings and underground mines, are ideal candidates for using a topological representation.

The major weakness of traditional skeletonization algorithms is that they suffer from high sensitivity to noise (perturbations of the boundary data). A small perturbation in the sensed data can drastically change the skeleton structure and its abstraction. This has led to

4.1 Introduction

Figure 4.1: The experimental platform used, a Turtlebot 2, with a Hokuyo laser range finder.

several different results on the stability of medial axis, including approaches which try to remove those skeletal branches that are likely to be generated due to boundary [27, 2, 26]. In the present work, we opt for AOF skeletons because they yield a rather direct and robust manner for finding skeletal points since the computation of AOF involves integrals rather than derivatives (see Figure 4.2). Here skeletal points are associated with locations where the average outward flux of the gradient of the Euclidean distance function through a shrinking circular neighborhood is non-zero. We shall later deal with changes to the topological structure of the skeleton due to boundary noise not by altering the representation itself, but by using a matching algorithm that employs spectral signatures.

The appeal of topological representations for mapping, exploration and human-robot interaction has been noted by several authors who suggested they be used directly, computed from 2D data [81, 25]. The Voronoi diagram, a classic structure in computational geometry that has appeared in many fields, and the Generalized Voronoi Graph (GVG) have been exploited in robotics as a mechanism for computing topological maps [29, 31].

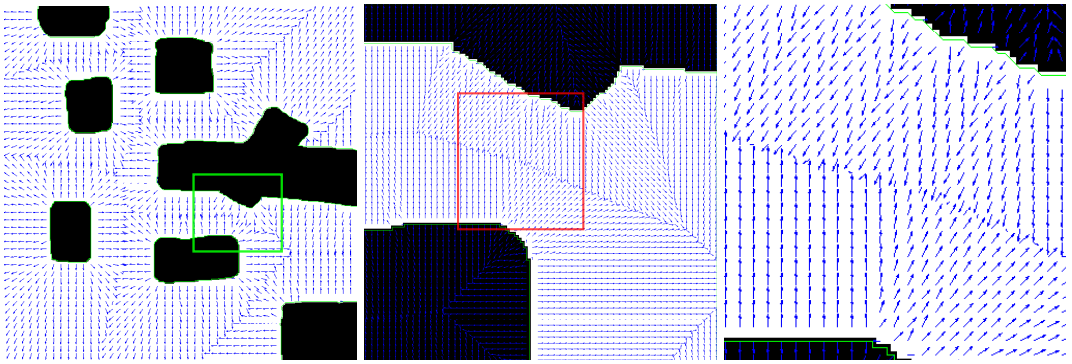


Figure 4.2: An illustration of the Euclidean distance function gradient vector field $\dot{\mathbf{q}}$ for a sample environment where the black regions represent obstacles. Computation of AOF which is based on the integral of the Euclidean distance function gradient vector field $\dot{\mathbf{q}}$ provides a stable skeleton computation (see Section 2.1).

Pure topological results have been studied in [40, 41, 42] for mapping a graph-like world with minimal sensor input. More recent work includes [152] on exploration strategies on a graph-like world.

The full employment of the GVG in a SLAM framework was proposed in [30], and extended for use in a hybrid metric/topological maps in [143, 88]. Tully et al. [145] recommend a hypothesis tree method for loop-closure where the branches that are considered to be unlikely based on topological and metric GVG information are pruned. Among the pruning tests, it is worth mentioning the planarity test which ensures that when a loop-closure has been decided, the resulting GVG graph remains planar. The utility of this test has been examined extensively in [125]. The purely topological variant of these approaches had been previously examined in [42].

Kuipers et al. [82] recommend the use of a framework, termed a hybrid spatial semantic hierarchy, where the incremental construction of topological large-scale maps is

employed in conjunction with metric SLAM methods for the creation of maps of small-scale. No use of a global frame of reference is made and a multi-hypothesis approach is used to represent potential loop-closures.

4.2 Mapping Environments using Average Outward Flux Skeletons

Our system takes laser scanned data from a 2D laser line scanner and generates an abstraction of the scanned environment. This is done through a number of modules: GMapping/binarization, average outward flux skeleton computation, pruning and simplification, and path planning for further exploration; see Figure 4.3. These modules are executed in a serial pipeline where the output of each module is the input to the next module.

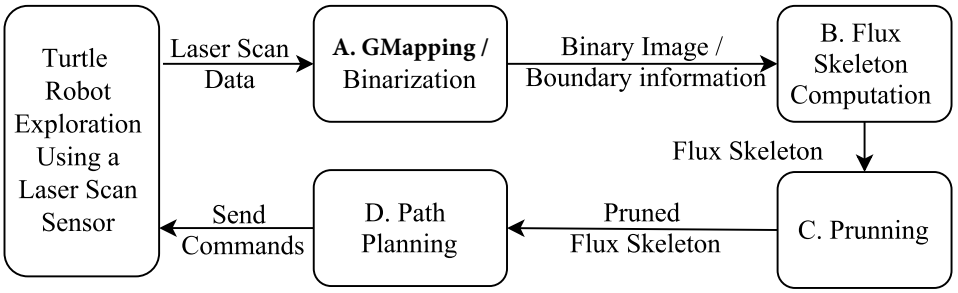


Figure 4.3: System overview. The system consists of four independently running modules along with a robot which is exploring the environment. Each of these modules is a component of a feedback chain system.

4.2.1 GMapping and Binarization

The system first receives 2D laser scan data in a format where each scan is a single line containing range measurements. These laser scan data serve as input to the GMapping module. GMapping is one of the most used laser-based SLAM algorithms [61]. It takes raw laser scan range data and odometry and produces gridmaps of the considered environment, where each gridmap is a probability distribution of cells (regions) being covered by the laser scan. The algorithm uses a highly efficient Rao-Blackwellized particle filter in which each particle has an individual map of the environment. The generated gridmap at the end of this stage is an intensity image where higher intensities show higher probabilities of being covered by the laser scanner (white regions), grey cells with lower intensities representing points which have not been covered yet by the robot, and where black cells usually represent walls where the range scanner has faced a physical obstacle. Figure 4.4(b) shows an example of a gridmap obtained after a certain amount of scanning.

Gridmaps must be binarized before they can be fed as input to our average outward flux-based skeletonization algorithm. To do this we apply the following sequence of steps: a) all pixels on gridmaps that are not scanned (gray regions) are set to background regions. Pixels that have a high probability of being obstacles (e.g. walls - black pixels) are stored as the foreground regions. b) Gaussian blurring is applied to smooth the structure that remains. c) The resulting image is then thresholded to give a binary one. d) The contours of all foreground regions in this image are extracted and sorted according to their area; regions having very small area are considered outliers. During this process, we keep track of the transformation needed to translate the final output to world coordinates. The second

row in Figure 4.4 depicts a binarized version of the grid map in the top row.

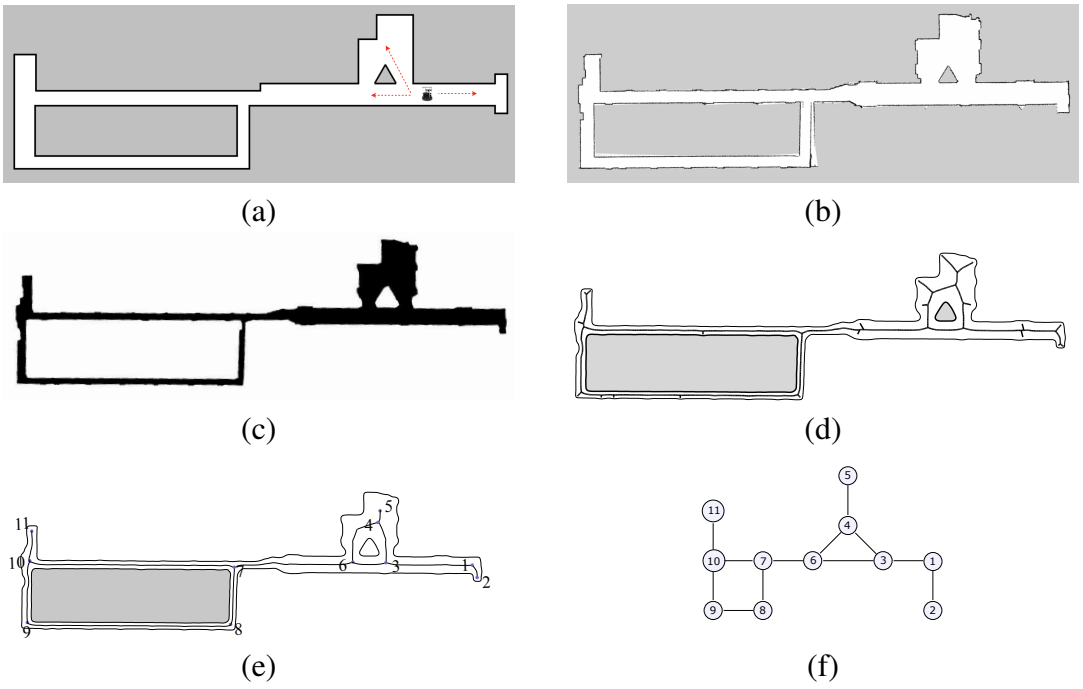


Figure 4.4: (a): An example map on an environment (b): The grid map of the environment. (c): the binarization of the grid map in the top row. (d): the full skeletonization process applied to the binarization of the environment. Although, the skeleton is very smooth, there are still branches that can be removed without altering its topology. (e): the skeleton in (d) is pruned and simplified in a way that makes robot navigation safe. Safe navigation means that robot does not go to an endpoint that is too close to a wall or an obstacle. (f): the topological map resulting from the abstraction in row four. Here, nodes are branch points or end points in the skeleton that are **not** removed by our pruning approach.

4.2.2 Pruning the Skeleton

As illustrated in Figure 4.4 (third row), the skeletonization process yields some branches that can be pruned without altering the topology of the skeleton. To prune such branches with the goal of topological mapping, we suggest a fairly simple but effective algorithm

Algorithm 4 Pruning Algorithm

```
1:  procedure ITERATIVE_PRUNING(Skeleton S, BinaryImage I)
2:      list: source_node ← ∅
3:      size ← S.nodes.SIZE
4:      for ∀ EndPoint E ∈ S do
5:          if IS_FEASIBLE(S,E) == false then
6:              S.nodes.REMOVE_END_POINT(E)
7:          end if
8:      end for
9:      if S.nodes.SIZE == size then                ▷ No reduction has been made
10:         return
11:     end if
12:     ITERATIVE_PRUNING(S,I)
13: end procedure
14:
15: procedure IS_FEASIBLE(EndPoint E, BinaryImage I)
16:     T ← EmptyImage
17:     τ ← a particular safety distance value for the robot
18:     τ′ ← a particular threshold on the area coverage
19:     for ∀ ContourPoint P ∈ I do
20:         if !P.IS_OBSTACLE & E.DISTANCE_TO(P) < τ then
21:             T.ADD_POINT(P)
22:         end if
23:     end for
24:     C ← T.GET_ALL_CONTOURS
25:     for ∀ c ∈ C do
26:         if c.AREA > τ′ then
27:             return true
28:         end if
29:     end for
30:     return false
31: end procedure
```

where the robot explores unseen regions and avoids getting too close to obstacles. Algorithm 4 summarizes this process which works as follows: it looks for all branches that have one branch point connected to an endpoint. If that endpoint is surrounded by walls

of obstacles then there is no possibility for further exploration in that direction. In such a case, the branch connecting the branch point to the endpoint is removed and the skeleton is simplified.

4.3 Path Planning

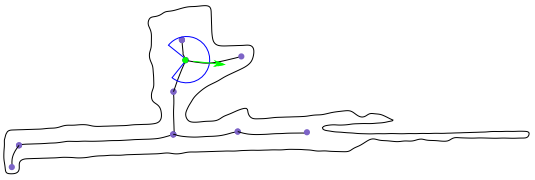


Figure 4.5: The environment has been partially explored and the robot now selects an edge (green) leading into unexplored space. The Pacman shape represents the current position and direction of the heading of the exploring robot.

The last step of this process is to guide the robot through the environment to explore new territory. As we showed in subsection 4.2.2, at each time, there is a partial abstraction of the environment generated by utilizing the GMapping package, discretizing the output map of GMapping, and then extracting the AOF skeleton. These steps produce an up to date topological map of the environment at each time step. The system keeps track of the visited nodes in a list. This enables the system to explore novel territory for its next move. As mentioned before, at each time step the robot moves from one of the visited nodes to a frontier node; the new nodes traversed on-route to the frontier node are added to the list of visited nodes. Each connecting edge is weighted by the length of the path through the skeleton. This weighting strategy results in a selection of good candidates for future exploration. In the algorithm, the nearest frontier node to the current node is selected. To

Figure 4.6: The exploring robot situated at one of the corridors of the McConnell Engineering Building at McGill University. This image is taken at the junction next to the triangular obstacle in the center of the map; see Figure 4.4.

find the nearest node, we use the *Bellman Ford* algorithm [7] which computes shortest paths from a single node to all of the other nodes in a weighted directed graph; see Figure 4.5.

4.4 Environment Mapping Experiments

Several experiments were performed, both in simulation and with a real robot. The proposed methodology was implemented under the ROS framework[1].

4.4.1 Experiments with a Real Robot

During the non-simulation experiments the Turtlebot 2 platform was used with a Hokuyo laser range finder; see Figure 4.6. The laser sensor has a range of 30 meters and has a 270° field of view, returning a dense cloud of 1080 coplanar points.

[1] http://www.ros.org/

Figure 4.7 presents the proposed algorithm in action using the Turtlebot 2 robot within the corridors of a floor in the McConnell Engineering building of McGill University. We emphasize that the scale of the map changes as the explored environment grows. The robot starts with a very limited view of the environment and the resulting skeleton is a simple curve, the concave part results from the limited field of view of the laser sensor; see Figure 4.7a. The robot identifies one side as a dead-end and proceeds down the corridor; see Figure 4.7b, until it detects a junction; see Figure 4.7c where the robot decides to follow the right side. Figure 4.7d shows the robot closing a loop, and then continuing down the corridor selecting the left edge, based on proximity; see Figure 4.7e. Finally Figure 4.7f presents the completed map of the environment.

4.4.2 Experiments with a Simulator

During the simulated experiments, the Stage simulator was used with a different environment, the cave world, as illustrated in Figure 4.8. The robot started at the middle of the environment, and created a skeleton based on the current information it had (Figure 4.8a); after moving to the nearest frontier node, more of the environment became visible and the topological map was updated; see Figure 4.8b. The lower left obstacle was not fully mapped, however, enough information was available to produce a loop. In Figure 4.8c the robot proceeded to explore the top left corner and then continue the exploration toward the branching nodes at the right side of the environment; see Figure 4.8d,e. Finally, the robot finished with a complete topological map of the environment in Figure 4.8f. The robot determines when the navigation through the environment is complete by checking the list of all visited nodes and unexplored nodes on the skeleton.

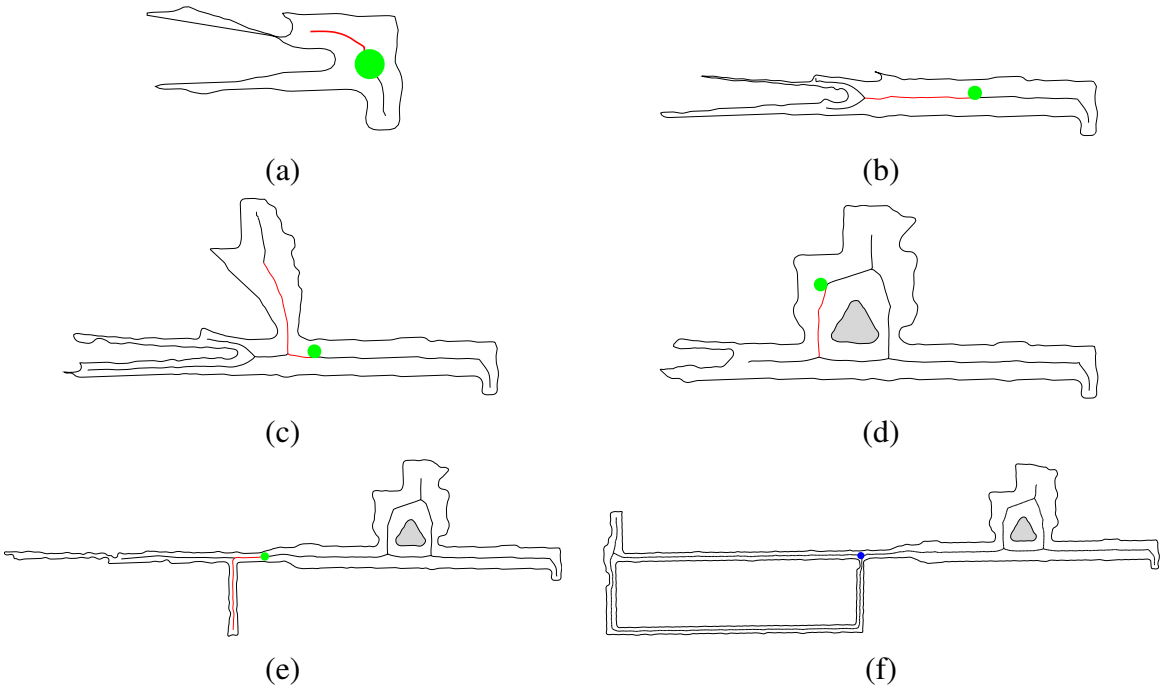


Figure 4.7: Six snapshots from an exploration in the corridors of the McConnell Engineering Building at McGill University's buildings. The experiment was conducted using the Turtlebot 2 robot. Similar to Figure 4.8, the green disk indicates the position of the robot and the red line the selected trajectory. The blue disk indicates successful construction of the skeleton-based map which shows when all the nodes in the topology map are visited.

4.5 Topology Matching

There are several scenarios which require the matching of two topological maps that are extracted from the same environment in robotics, including map merging, place detection, and map evaluation. One of the ways to evaluate the robustness of an autonomous environment mapping algorithm is to start from different locations on the same environment and then compare the topology maps with each other. If the resulting topology maps are

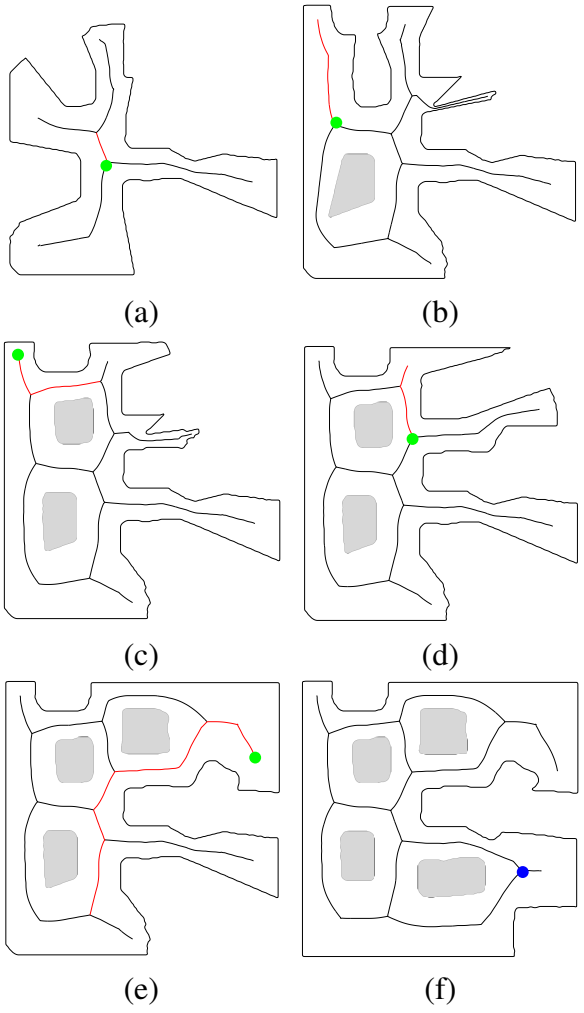


Figure 4.8: Six steps of the exploration algorithm, using the Stage cave simulated world, are shown here. At each step, the robot's position, the skeleton of the mapped environment, obstacles, and the future path is shown. The green disk represents the robot, and the red path is where the robot will traverse next. (f) The pose of the robot is drawn in blue to indicate that the robot has now fully explored the map.

not similar enough, the environment mapping algorithm needs to be revisited in terms of stability of the representation. Sometimes, an instability in an autonomous algorithm, like ours, can arise from an outside module that has been used in the system. In our case,

our environment mapping algorithm can fail if the GMapping module fails. The question of whether our algorithm can recover from these challenging situations then arises. To analyze such failures cases, we evaluate how well areas from one map partially match to areas of another map, i.e., how well the robot can know it has successfully visited an explored area. For these cases, we develop a method that can match two topological maps using the AOF skeleton of their environment, while permitting structural alterations due to perturbations in the sensed boundary.

To design a matching algorithm for topology maps, we propose using the spectral correspondence algorithm of Lombaert et al. [90], designed originally for the task of dense vertex correspondence between two surface meshes. In their work, they present an approach to dense vertex-to-vertex correspondence that uses direct matching of features defined on a surface, and then improve it by using spectral correspondence as a regularization. Applications include finding correspondences between meshes undergoing non-rigid transformations and articulated meshes. The algorithm can also be used for precise and accurate correspondence in medical imaging. Representing a structure like a topology map as a graph, a spectral correspondence algorithm tries to characterize that graph via its spectrum, which is the set of eigenvalues and eigenvectors of either the adjacency matrix or the closely related Laplacian matrix. We can think of different ways to represent a matching task as a spectral correspondence process.

In the following, we will discuss how topology maps are arranged in the form of graph representations and used for spectral correspondence. First, let us assume we have two topology maps and nodes and their connectivities in these maps are represented in the

form of a graph. Each skeletal point on the AOF skeleton can be represented by a 3-tuple (x, y, r) where (x, y) represents the skeletal point location and r represents the radius of the corresponding inscribed disk at that particular point. Assume that these two graphs are $G_1 = \{V_1, E_1\}$ and $G_2 = \{V_2, E_2\}$. Note that the correspondence algorithm of Lombaert et al. [90] does not require that $|V_1| = |V_2|$ or $|E_1| = |E_2|$. The mapping of environments using the FOCUSR algorithm can be described via a four-stage process. What is different here is that rather than choose as an input graph the nodes of a mesh representing the boundary of a 2D or 3D surface, in the current context we shall use the graph based on the AOF skeleton as an input.

1. **Computing spectra**: In our first configuration setup, a graph is made from the topology map extracted from AOF skeletal points and the connectivity of the nodes is based on whether the medial points are neighboring (in which case we place a value of 1 in the corresponding entry in the adjacency matrix) or not (in which case we place a value of 0 in the adjacency matrix). Having the graph, its adjacency matrix is derived based on the connectivity of nodes in the topology map. In a discrete version of the medial axis as a set of pixels on a grid, two nodes are neighbors if they share either an edge or a corner point. Grady et al. [60] formulated the general Laplacian operator as:

$$\mathcal{L} = G^{-1}(D - W), \tag{4.1}$$

where W is a weighted adjacency matrix of the graph with affinity weights (see [60]). We can consider different metrics for the weighted adjacency matrix W. In our implementation, we have considered two different metrics for each entry of W,

w_{ij}: a) , $w_{ij} = \frac{1}{dist(i,j)}$, where this value is also multiplied by the connectivity link value between nodes i and j (1 when they are neighbors and 0 when they are not), and b) $w_{ij} = e^{\frac{-dist(i,j)^2}{2*\sigma^2}}$ where like case a) we consider the connectivity link between nodes also. In our first configuration, the distance between two nodes v_i and v_j is computed as:

$$dist(i,j) = \|(\mathbf{p}_i, \gamma F_i) - (\mathbf{p}_j, \gamma F_j)\|_2 \tag{4.2}$$

where $(\mathbf{p}, \gamma F)$ is the concatenation of the 2D coordinate values $\mathbf{p} = (x, y)^T$ with the K feature values $F = (f^{(1)}, ..., f^{(K)})^T$. γ is a $K \times K$ diagonal matrix which contains the K weights controlling the influence of each feature. When there is only one feature, the matrix γ reduces to a single number. In our configuration, for every medial point, we consider the radius of the maximal inscribed disk. This feature can play a very important role in matching because thicker (wider) regions get a larger weight in terms of their influence on the matching score. The degree matrix, D, is a diagonal matrix, where $D_{ii} = \Sigma_j W_{ij}$, and G can be any meaningful node weighting matrix where nodes with significant features are more heavily favored in the match.

2. **Reorder and align spectra**: When spectra are computed, two situations are possible that make the direct comparison of spectral coordinates challenging. First, computing eigenvectors may generate a sign ambiguity. Second, it is possible that when eigenvectors are being computed for the same value in two maps they might be computed in opposite orders due to the fact that the ordering of the lowest eigenvector may change. Lombaert et al. [90] suggest mitigating the effects of the flipping problem by favoring three factors: 1) pairs of eigenvectors that are most likely to

match based on the similarity between their eigenvalues 2) histograms 3) the spatial distributions of their spectral coordinate value. The process of reordering is sped up by downsampling all eigenvectors. In our experiments, since we are considering points of a 2D medial axis in our configuration (which are far fewer in number than the number of points on a typical 3D surface mesh) we can consider most of the eigenvectors without worrying about the cost of reordering.

3. **Find matches**: After reordering and aligning the spectra, two points which are closest in the embedded representations could be treated as corresponding points in both topology maps. This is achieved by using the Coherent Point Drift (CPD) method [100].

4.6 Experiments with Topology Mapping and Matching

For this project, historically, the initial implementation of the AOF skeleton code was written in C and the stage simulator was used to test the autonomous navigation of the environment. These implementations were the basis of the results published in [116]. For the topology matching extension, we used a different setup. First, a virtual machine with ROS Hydro and Gazebo for the Robotics System Toolbox™ was installed on our system. This allowed us to have a ROS Hydro Desktop installation, a Gazebo robot simulator 1.9.6, and some sample Gazebo worlds for a simulated TurtleBot. This allowed us to connect to the virtual machine through its network IP address. This turned out to be extremely useful because we could implement all the other steps in Matlab (see Figure 4.9).

To be able to carry out experiments, we ran examples of simulated maps using Gazebo

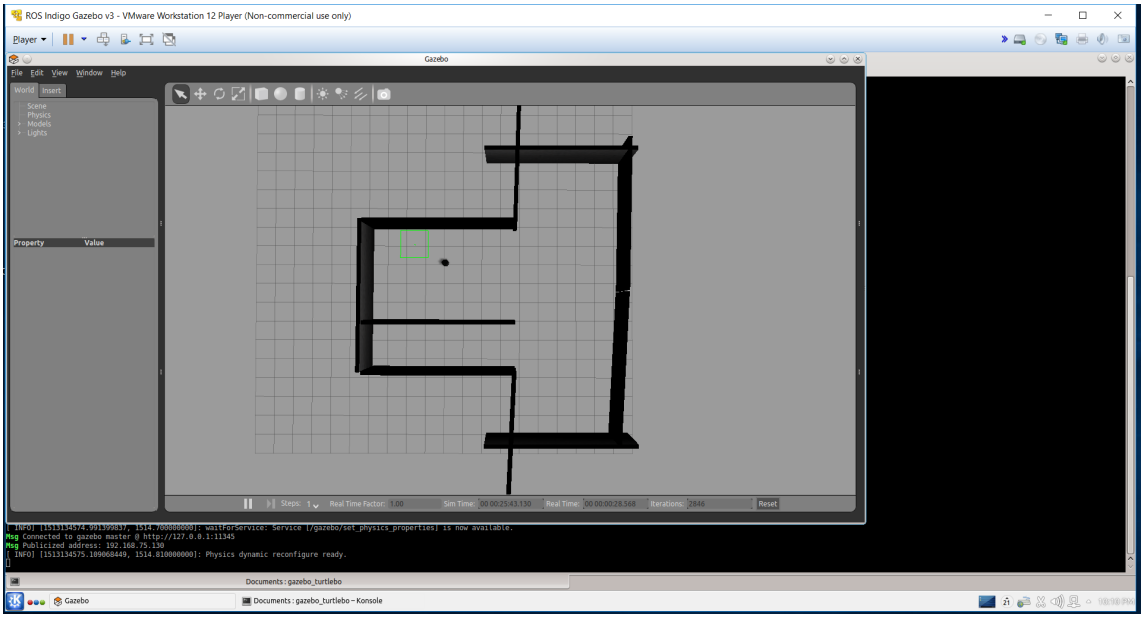


Figure 4.9: A screenshot of the virtual machine environment with the Gazebo simulator installed.

and the Turtlebot robot. For each environment map, we made a launch file with a specific map and a Turtlebot that has a laser scanner range finder installed on it.

Once the GMapping module was working, we carried out an online binarization of images captured from the occupancy grid by writing code in Matlab. We first cropped the occupancy grid map to the area that cells were occupied by values other than the background. Then the image intensities were placed into three categories: 1) scanned pixels 2) obstacles and walls, and 3) background. To lower the effect of the noise generated by the laser scan and GMapping(e.g. sharp rays), Gaussian blurring with a dynamically chosen smoothing scale was applied (see Figure 4.10). The occupancy grid image was then thresholded to give us a binary image (see Figure 4.11).

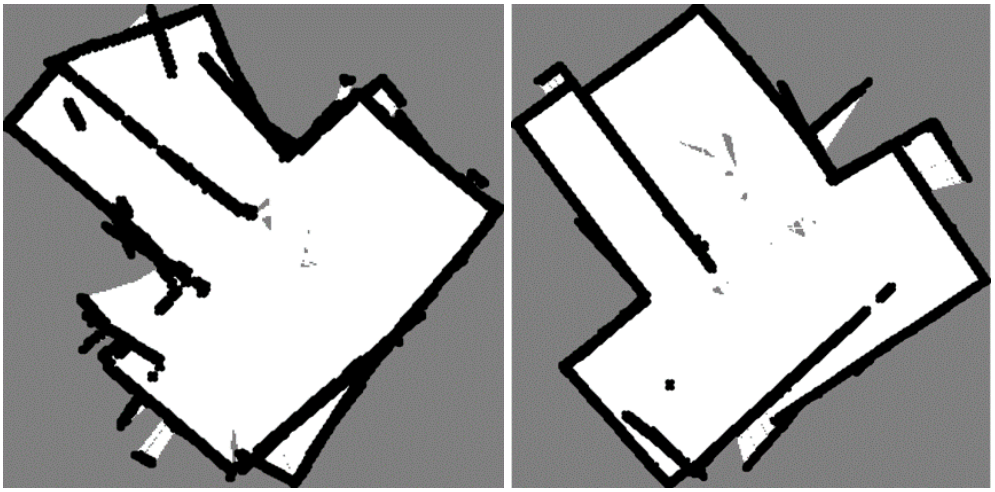


Figure 4.10: This figure shows two scans of a single environment where for each scan the robot started from a different location. As it can be seen, the left example was deliberately scanned carelessly just to test how robust the algorithm would be in such a circumstance.

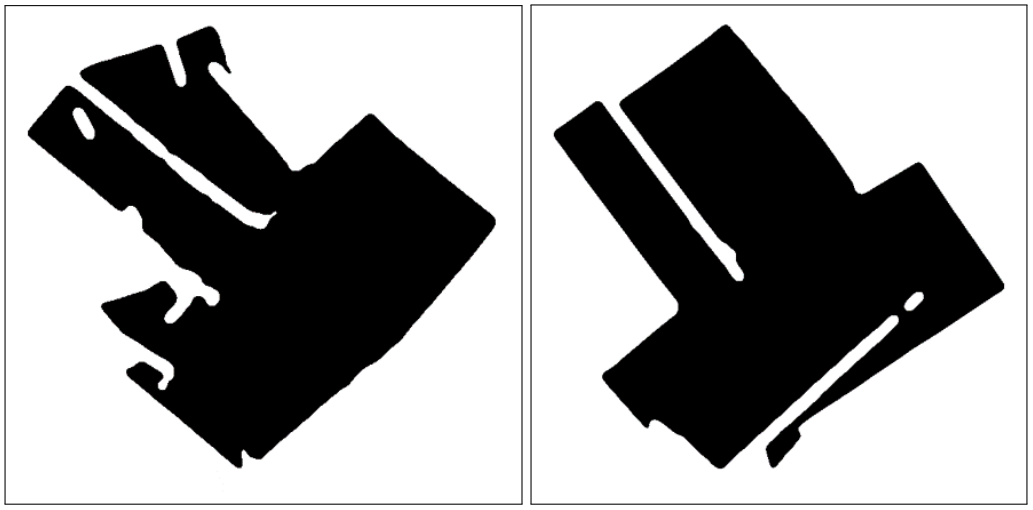


Figure 4.11: These two images show the result of the binarization process for the occupancy grids shown in Figure 4.10.

As illustrated in Figure 4.12, the skeletonization process produces branches that should be pruned without altering the skeleton's topology. To prune such branches with the goal

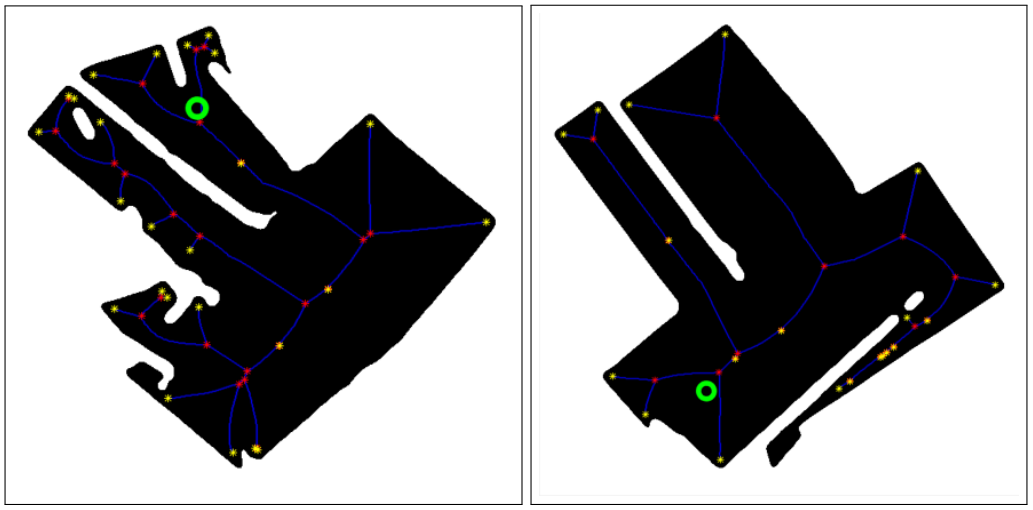


Figure 4.12: This figure shows the result of the skeletonization process on the binary images computed in Figure 4.11. The skeletal points are shown in blue here. Branch points are represented by red stars and endpoints are shown by yellow stars. The green disk in each image shows the location where the robot has started the environment mapping. As can be seen, it is not immediately obvious how these two different environment maps could be aligned. To be able to compare these environments, one must work at an appropriate level of abstraction of the skeleton, which is in effect the capability that spectral correspondence provides.

of topological mapping, we suggest a fairly simple but effective algorithm where the robot explores unseen regions and avoids getting too close to obstacles. Algorithm 4 summarizes this process.

4.6.1 FOCUSR Setup

To apply the FOCUSR algorithm, we used the MATLAB implementation for matching surfaces introduced in [90]. This method matches the meshes of 3D surfaces. We recoded the package for 2D medial axes by devising 2D medial graphs as follows. We consider $G = (V, E)$, where V represents all medial axis points and E represents their

connectivity based on their original connectivity in the skeleton of the environment. Each vertex of V is represented by a quadruple $\mathbf{p}_i = (x_i, y_i, r_i, \theta_i)$, representing the position in x-axis, the position in y-axis, the radius value at that point (which is the closest distance to the boundary point), and the object angle respectively. Notice that θ represents the object angle, which is expressed for the unit tangent in the direction of decreasing radius along the medial branch curve.

4.6.2 Results and Discussion

We tested the implemented algorithm on two environment maps and we achieved promising results. The visualization shows results that are plausible, i.e., qualitatively similar regions of the maps in terms of spatial layout and local width, seem to align (see Figure 4.13). In addition to the previous example, where the robot starts from 2 different loca-

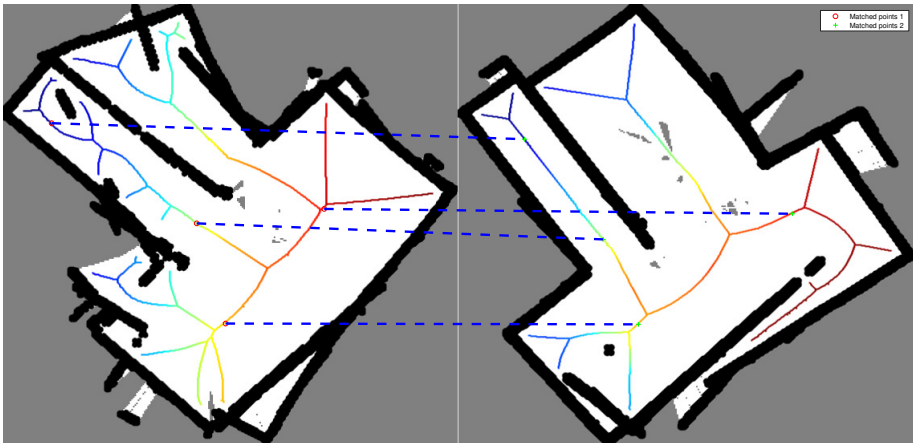
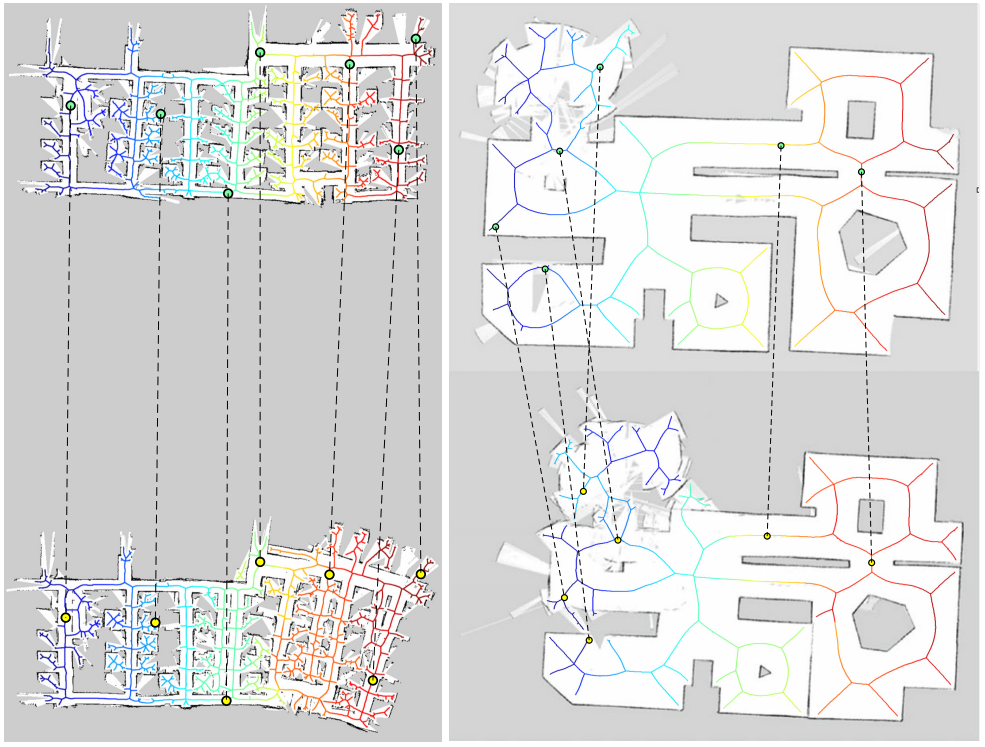


Figure 4.13: This figure shows the correspondence map between two computed topology graphs, based on their spectral correspondences, with corresponding points shown with similar colors.

tions on the map (Figure 4.12), we tried other environments where the GMapping module

partially fails in mapping the environment, resulting occupancy in grid maps that do match exactly to the scenario where the environment is correctly sensed. To see if the maps generated from these situations can still be matched, we present the result of our topology mapping for two additional environments in Figure 4.14. The results show qualitatively plausible matches in terms of the correspondences found between branches of the main (the widest) regions. One of these examples, the one to in Figure 4.14 (a), is considerably more complex in terms of size and topology than the other examples considered in this chapter.



(a) (b)

Figure 4.14: Same environments mapped differently matched against each other.

Although our algorithm provides some plausible examples of how two environments

that are mapped differently can match to each other, we acknowledge that there exists a challenge in evaluating this method quantitatively, and we provide some numbers that put this to perspective. One potential way to quantify this matching is to look at the path similarity between endpoints of the graphs computed from each environment. When correspondences are found between two mapped environments, we can compare each path that connects each two endpoints from one environment to their corresponding ones from another environment. Let us assume we have two explored environments and their medial axes are represented as G and G', and the mapping function that maps the correspondences between them is represented by $\mathcal{T}(\mathbf{p}_i) = \mathbf{p'}_j$, where $\mathbf{p}_i \in G$ and $\mathbf{p}'_j \in G'$. If we let $\mathbf{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and $\mathbf{E}' = \{\mathbf{e}'_1, \ldots, \mathbf{e}'_m\}$ represent the end points in G and G' respectively, we may then present the following distance measure between these environments as:

$$d(G, G') = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} pd(p(\mathbf{e}_i, \mathbf{e}_j), p(\mathcal{T}(\mathbf{e}_i), \mathcal{T}(\mathbf{e}_j)}{n(n-1)} + \frac{\sum_{i=1}^{m} \sum_{j=i+1}^{m} pd(p(\mathbf{e'}_i, \mathbf{e'}_j), p(\mathcal{T}(\mathbf{e'}_i), \mathcal{T}(\mathbf{e'}_j)}{m(m-1)} \tag{4.3}$$

where, $p(\mathbf{e}_i, \mathbf{e}_j)$ represents the path between two endpoints $\mathbf{e}_i$ and $\mathbf{e}_j$ in the respective graph (where those nodes belong to) and $pd(\text{path}_1, \text{path}_2)$ represents an elastic matching between two paths. To compute an elastic matching between two paths, we can use the Optimal Subsequence Bijection method, first presented in [162], that accepts two finite sequences of end nodes of skeletons and finds the best possible correspondences between

them by using a cost function that measures how much those end to end points paths are similar to each other.

4.7 Discussion

A new methodology for the exploration and mapping of an unknown 2D environment was presented in this chapter. The algorithm belongs to the family of sensor-based topological maps. It would be useful to carry out a comparison between the use of AOF skeletons, and the use of Voronoi skeletons, as is popular in the literature on environment mapping ([55],[31], and [32]). Carrying out a detailed analysis is beyond the current scope of the work reported here. But it is worth pointing out that the AOF skeletons demonstrate some robustness to a degree of boundary perturbation, an issue that can plague topological representations computed using traditional Voronoi approaches. Utilizing all the recorded data up to the current step results in efficient loop closures and the elimination of the side effects of noise. Experimental results from synthetic as well as live data from an exploring robot demonstrated the efficiency and robustness of the proposed framework. In addition, we proposed a novel spectral correspondence based matching algorithm between topology maps of environments using AOF skeletons. Experiments show that our algorithm produces promising results, in terms of finding correspondences between similar regions, despite alterations to the graph structures themselves due to simulated sensor noise.

Future extensions of this work could consider the adaptation of the motion planning technique to deploy on aerial vehicles, such as quadrotors, where the smoothness of the trajectory would be of paramount importance.

Part III

Scene Categorization : Medial Axis Based Measures

5

Scene Categorization by Human Observers

The contents of this chapter are largely based on the article "Local contour symmetry facilitates scene categorization" [159] which grew out of a collaboration with colleagues in the human and computer vision groups at the University of Toronto. I was solely responsible for the average outward flux-based medial axes, scores, and algorithms for scene symmetry. In addition, I was primarily responsible for the preparation of the splits and contributed to the analysis along with many other aspects of this research project. This work was carried out under the academic supervision of Prof. Siddiqi.

People are able to rapidly categorize briefly flashed images of real-world environments, even when they are reduced to line drawings. This setting allows for the study of time-limited perceptual grouping processes in the human visual system that are applicable

to line drawings. Previous work [158] showed that standard local features of individual contours, or junctions between contours, do not account for this rapid classification ability but, rather, the relative placement of these contours appeared to be important. Here we provide strong support for this observation by demonstrating that local ribbon symmetry between neighboring pairs of contours facilitates the categorization of complex real-world environments. To this end, we introduce a novel computational approach, based on the medial axis transform, for measuring the degree of local ribbon symmetry in a line drawing. We use this measure to separate the contour pixels for a given scene into the most ribbon symmetric half and the least ribbon symmetric half. We then show human observers the resulting half-images in a rapid-categorization experiment. Our results demonstrate that local ribbon symmetry facilitates the categorization of complex real-world environments. This is the first study of the role of local symmetry in inter-contour grouping for human scene classification. We conclude that local ribbon symmetry appears to play an important role in jump-starting the grouping of image content into meaningful units, even in flashed presentations.

5.1 Introduction

Gestalt grouping rules, such as good continuation, symmetry, or similarity, were proposed as a qualitative account for how edge segments or shape parts are grouped into larger structures [77, 154, 79, 78]. However, there is so far no mechanistic, quantitative model of how Gestalt rules are implemented and used to facilitate the visual perception of complex real-world scenes.

5.1 Introduction

Following its postulation as one of the Gestalt laws of perceptual organization, symmetry has been investigated as a grouping principle in both human and computer vision [78, 69, 147, 51, 140, 89, 149, 109]. We define symmetry as a redundancy in the shape of an object or its projection onto the image plane due to a similarity between sub-pieces of a larger part. In the context of an image, this can include mirror-symmetry, where part of the image is reflected across an axis, rotational symmetry, where a section of the image is a copy of another section but at a different orientation, as well as translational symmetry, where a section of an image is a translated copy of another section. These forms of symmetry can either apply to part of an image (local symmetries) or to the entire image (global symmetry). Local symmetries do not need to apply to an entire object. In fact, a single part of an object may be locally symmetric. For example, consider a building with Greek columns. If the building is viewed from an oblique angle, the projection of the building onto the image plane does not necessarily result in a symmetric image. However, the projection of a single pillar in this view may still be locally symmetric.

We can consider many different types of local symmetry. Medial symmetry applies to those types of local symmetry that are the result of a reflection across a curved axis. This is a type of mirror-symmetry on a local scale. The medial axis transform provides a way to capture medial symmetry [18]. The intuitive idea behind medial symmetry is that the boundary of a shape can be formed by sweeping a disk along a suitable path (the medial axis).[1] As a special case of medial symmetry, ribbon symmetry occurs when the radii of the medial disks remain constant along the axis. Parallel lines are one example of ribbon

[1] A closely related representation in 3-D is Binford's *generalized cylinder* which sweeps a 2-D cross-section along a 3-D space curve [13].

symmetry. Another example is a river of constant width, which winds through a field. In this chapter, we restrict our consideration to ribbon symmetry.

Since first theorized as a grouping principle by the Gestalt psychologists, there has been a long history of research on symmetry (global symmetry) and its influence on human vision (for reviews, see [148] and [149]). In addition to global symmetry, local symmetry influences several aspects of human vision [96, 80, 21, 51, 24, 85, 54]. Machilsen et al. [93] showed that mirror-symmetric shapes are easier to detect than asymmetric shapes when embedded in a noise field. Wilder et al. [157] showed that medial symmetry also plays a role in object detection, where shapes with a simpler medial axis structure were more easily detected. The medial axis also helps explain human shape categorization [156]. This work is important for the current study as it demonstrates that symmetry has an influence on visual processing both prior to object detection and post-detection during classification. As images of real-world scenes are rarely globally symmetric, we will focus on local symmetry in the current study. Specifically, we will base our approach on the medial axis. While some previous work computes medial axes only on closed object silhouettes, we here apply our consideration of local symmetry beyond individual objects through a computational approach which is applicable to entire scenes.

Rapid scene perception does not require color photographs; observers can rapidly classify line drawings of real-world scenes [12, 150]. Furthermore, recent work demonstrates that scene content is carried primarily in the high spatial frequencies [9]. In fact, the high-pass images used in the latter study closely resemble line drawings. Additionally, Walther et al. [151] found that photographs and line drawings result in similar neural pat-

terns, showing that the underlying category-specific representations are similar. Thus, we choose to use line drawings, without fear of loss of generality of our results, in order to allow us to study the influence of shape alone, without the confounds introduced by color and texture.

Various properties of the contours that are preserved in line drawings have been assessed for their role in scene perception. In a recent study, using an experimental design similar to the present study, the role of contour junctions in rapid scene classification was directly tested [158]. The results showed that scenes from which the junctions were removed were more easily classified than scenes form which the middle sections, between junctions, were removed. The local relationships between elongated sections play at least as important of a role in scene perception, as opposed to the intersection between contours, hinting at the importance of local symmetry relationships. Since symmetry was not directly measured or manipulated in [158], conclusions about its importance for scene classification were not able to be drawn. Here we explicitly measure local ribbon symmetry in complex, real-world scenes and test for its importance for scene categorization.

We measure local ribbon symmetry by extracting the symmetric axes from line drawings of entire real-world scenes. Symmetric axis representations are often defined mainly for individual object silhouettes [18, 136, for example]. Here we apply the concept of a symmetric axis to entire scenes. We use the symmetric axis to assign a symmetry score to each contour pixel in an image. We then split the image into two halves, one containing the high-symmetry half and the other containing the low-symmetry half of the pixels. The two half-images have no contour pixels in common and, when combined, result in the

original, intact line drawing. We use both versions, along with the intact line drawings, in a categorization experiment. If symmetry is indeed a strong cue for scene processing, then the symmetric half-images should be more easily classified than the asymmetric half-images.

5.2 Methods

5.2.1 Scoring Symmetry

In order to measure the degree of ribbon symmetry present along individual contours in a line drawing of a scene, we have devised a measure of symmetry-based upon AOF skeletons (see Section 2.1). The method described in the present section is the first measure used in our work to characterize ribbon symmetry using medial axes. Later, in Chapter 6, we shall introduce other ratio-based medial axis measures for describing contour ribbon symmetry, taper symmetry and local contour separation.

To begin, we explain how the AOF Skeletonization approach is applied in our pipeline. Our illustrative example is based upon the line drawing in Figure 5.1 (top right). We consider the line drawing of a scene as a binary image with contour (black) pixels and non-contour (white) pixels. Each region inscribed by a subset of line drawings segments and/or the boundary of the image is considered an input shape for our AOF skeletonization approach. As discussed in Chapter 2.1, each non-contour point in the scene is assigned its Euclidean distance to the closest contour point (see Figure 5.1 (c), here, color represents the distance to the nearest contour: blue is a small distance, and red is a large distance), and

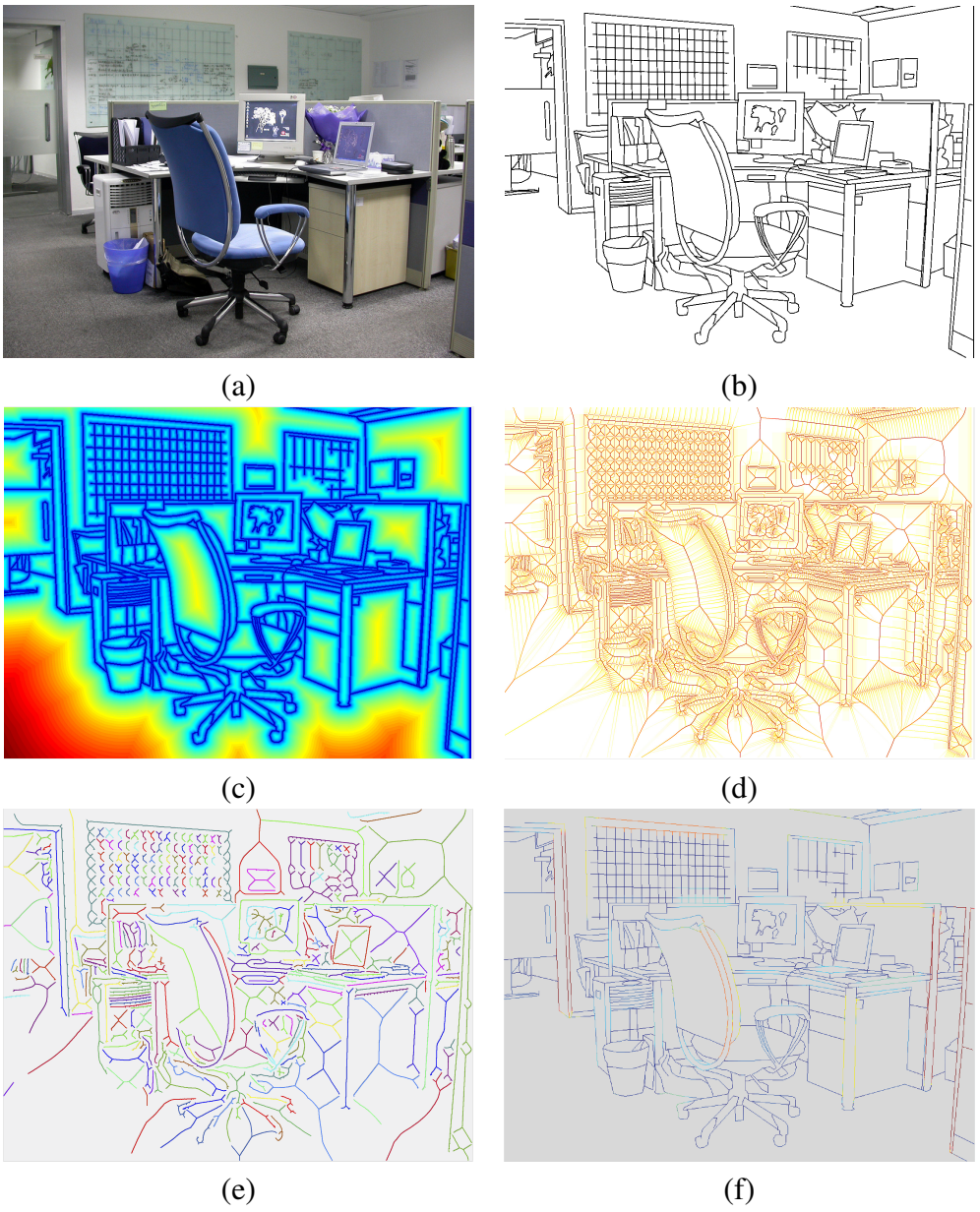


Figure 5.1: A photograph of an office scene (a), along with its artist-traced line drawing (b), outward distance transform (v), average outward flux (AOF) map (d), flux skeletons (e), and symmetry score at each contour pixel (f).

once we have the Euclidean distance function, we compute its gradient, and the outward flux of the gradient, through a shrinking disk placed at each non-contour point normalized by the perimeter of the disk. By thresholding the AOF (see Figure 5.1 (d)) at each point, a set of skeletal branches for each region in the image is obtained (see Figure 5.1(e), here, the color simply denotes skeletons for different closed regions).

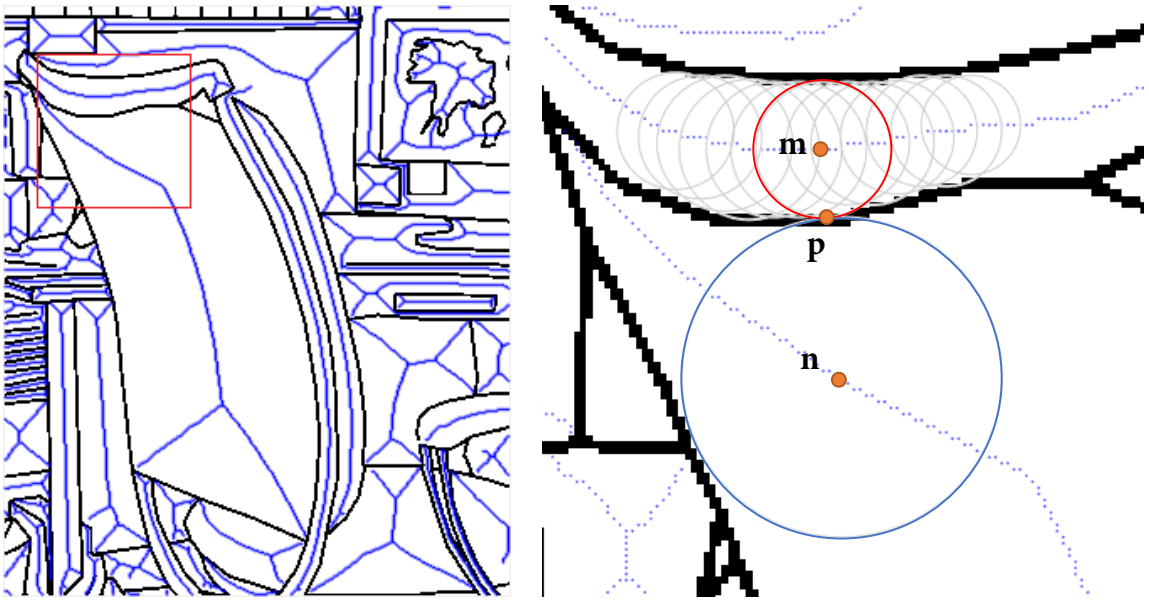
The next step is to assign a score of symmetry to each point on the line drawing. This is done by first scoring points on the medial axes, and then transferring these scores to the contour pixels. A medial axis point is given a symmetry score related to the degree of local parallelism between the contours on either side of the medial point. The specific score we use is equal to the fraction of medial points in a local neighborhood for which the derivative of the radius function is below a set threshold. This process is illustrated for the medial point $\mathbf{m}$ shown in Figure 5.2, where neighboring medial points are illustrated with lightly shaded circles. The radii of these circles are a slowly varying function of the position along this medial branch and therefore (depending on the threshold used for the derivative) we might expect $\mathbf{m}$ to have a high symmetry score.

Once the scores have been computed for all medial axis points, we then map these scores to points on the boundary contours by noting that each point on the boundary is associated with two skeletal points, one on each side of the contour. This is illustrated at the boundary point $\mathbf{p}$ in Figure 5.2, which is associated with the two medial axis points $\mathbf{m}$, and $\mathbf{n}$ on either side (that is, the boundary point $\mathbf{p}$ provides an active constraint on the size of the disks centered on the medial axis points $\mathbf{m}$ and $\mathbf{n}$). The score at $\mathbf{p}$ is then defined to be the maximum of the scores at the two associated medial axis points $\mathbf{m}$ and $\mathbf{n}$. Taking

the maximum makes intuitive sense because the boundaries belonging to an object are non-accidentally related and are more likely to be in a ribbon symmetric relationship than are the boundaries of that object with other structures. An object boundary and a boundary in the background (or the boundary of a different object) are only parallel if they are accidentally aligned. In our example, we would expect the score at $\mathbf{m}$ to be larger than the score at $\mathbf{n}$, since the radius function at $\mathbf{n}$ is changing more rapidly, and this process would assign the symmetry score at $\mathbf{m}$ to the boundary point $\mathbf{p}$. This provides a measure of the local parallelism of the boundary in the neighborhood of $\mathbf{p}$ with neighboring boundary points on one or the other side of that contour fragment (see Figure 5.1 (f), here, blue represents a weaker symmetry score and red represents a stronger symmetry score). Note that pairs of long smooth parallel contours, such as down the side of the chair, receive a large symmetry score. Non-parallel regions receive low scores. Square regions also receive low scores, because the medial axis is influenced by all four sides of the region, not just two parallel sides. This procedure is detailed in Algorithms 5 and 6.

Algorithm 5 Scoring Skeletal Points

procedure SYMMETRYSCOREFORSKELETALPOINT(Skeletal Point $\mathbf{m}$)
    Consider a window of $2K+1$ skeletal points centered at $\mathbf{m}$.
    Let us assume these $2K+1$ points are $\mathbf{m}_{-K}, ..., \mathbf{m}_K$, where $\mathbf{m}_0 = \mathbf{m}$
    $r(\mathbf{m}_i) \leftarrow$ the radius value of the maximal inscribed disk centered at $\mathbf{m}_i$
    $\tau \leftarrow$ a particular marginal threshold
    Assign the score of symmetry as:
    $\mathcal{S}(\mathbf{m}) = \frac{\#\{\mathbf{m}_i | \forall_{i=\{-K,\ldots,K-1\}} \text{ where } \frac{|r(\mathbf{m}_i)-r(\mathbf{m}_{i+1})|}{\max(r(\mathbf{m}_i),r(\mathbf{m}_{i+1}))} <= \tau\}}{2K}$
    return $\mathcal{S}(\mathbf{m})$
end procedure
▷ Intuitively, $\mathcal{S}(\mathbf{m})$ represents the fraction differences between of adjacent disk radii that are small, i.e., their boundaries on either side are close to parallel.



(a) (b)

Figure 5.2: (a) Using a portion of the office scene in Figure 1, around the back of the chair, (b) we illustrate the manner in which a contour point $\mathbf{p}$ is given a symmetry rating. The boundary point is associated with two skeletal points on either side, $\mathbf{m}$ and $\mathbf{n}$. In the vicinity of each such skeletal point, the variation of the radius function is used to assign a symmetry score, as described in Algorithm 5. The grey circles depict the maximal inscribed disks along with the interval under consideration around $\mathbf{m}$. The point $\mathbf{p}$ receives the larger of its two symmetry scores.

Having designed a method for scoring symmetry of line drawings, we can apply it to our database of lines drawings. The line drawings we used were first described in [151]. Each line drawing was obtained by having a photograph traced by an artist, who was given the instruction:

> For every image, please annotate all important and salient lines, including closed loops (e.g., the boundary of a monitor) and open lines (e.g., bound-

Algorithm 6 Scoring Contour Points

procedure SYMMETRYSCOREFORALLLINEDRAWINGPOINTS
  for each line drawing point $\mathbf{p}$ **do**
    Find the centers of the inscribed disks that touch point $\mathbf{p}$
    Let us assume these centers are called $\mathbf{m}$ and $\mathbf{n}$
    $\mathcal{S}(\mathbf{p}) = \max(\mathcal{S}(\mathbf{m}), \mathcal{S}(\mathbf{n}))$
    return $\mathcal{S}(\mathbf{p})$
  end for
end procedure
  ▷ In the generic case two maximal disks touch point $\mathbf{p}$, one disk from each side of $\mathbf{p}$, however there are cases in which more than two disks touch $\mathbf{p}$. In those cases, take the max score over all disks.

aries of a road). Our requirement is that, by looking only at the annotated line drawings, a human observer can recognize the scene and salient objects within the image.

We used 72 images from each of the six categories (beaches, forests, mountains, city streets, highways, and offices). After scoring the symmetry in each image, we analyzed the distribution of symmetry scores by category. For each category, the distribution is skewed toward low-symmetry pixels. Some categories (e.g., beaches) have relatively more low symmetry scores than others (e.g., mountains), which have few low symmetry scores. Category differences are most easily apparent in the distributions of mean symmetry scores, see Figure 5.3. Here, we compute the average symmetry score for an image and record the distributions of averages for each category. These distributions reveal that the symmetry ratings do differ by category, and thus the symmetry values could be potentially used in the categorization of a scene. Cities and offices have the lowest average mean symmetry score (0.0725 and 0.102). Next are forests, with an average of 0.129. Then, mountains and

highways both have means of 0.141. Finally, beaches have the highest average symmetry (0.166). This may seem surprising, given that we think of human-made buildings and objects to be symmetric, but recall that we are specifically measuring ribbon symmetry. Our measure gives high scores to elongated regions. For example, in a beach scene, as waves roll in they tend to create pairs of ribbon symmetric lines. Many objects in an office or city, while symmetric in the real world, are not ribbon symmetric in a *2D* image due to perspective foreshortening.

5.2.2 Stimuli

Having established a new method for scoring ribbon symmetry, we selectively removed either the most or the least symmetric contour pixels of line drawings of natural scenes. The line drawings were the same drawings used in the study by Walther et al. [151], who obtained them by having artists trace photographs. We either showed the original line drawing (the intact condition), a line drawing with the most symmetric 50% of the contour pixels retained (the symmetric condition), or a line drawing with the least symmetric 50% of the contour pixels retained (the asymmetric condition). Example stimuli are shown in Figure 5.4 and 5.5. If ribbon symmetry is influential in scene processing, then we should expect performance to be better in the symmetric than the asymmetric condition.

5.2.3 Participants

Twenty-six undergraduate students in an introductory psychology course at the University of Toronto (19 female, 7 male, mean age 18.3) participated in the experiment for course

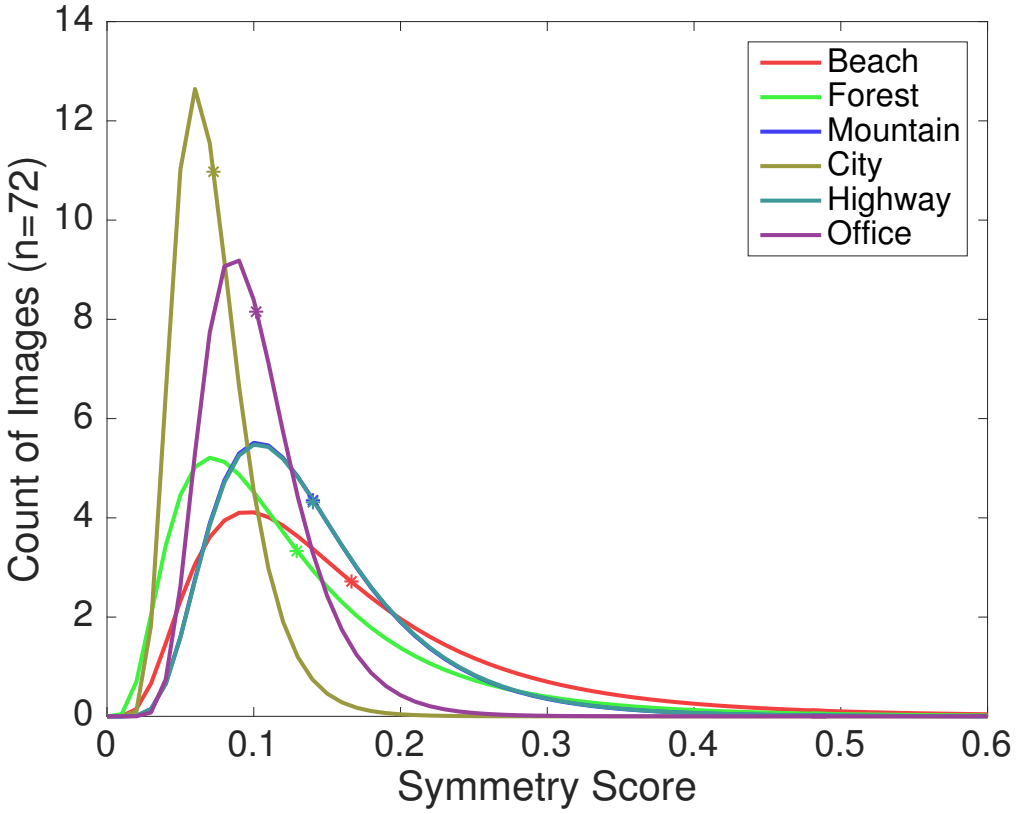


Figure 5.3: Distributions of average symmetry scores. Each distribution is composed of the mean symmetry score for each of the 72 images in that category. The distributions shown are fit using a log-normal distribution. The means are shown as the '*' symbol. Two distributions, Mountain and Highway, overlap, which is why it may appear as if there are only five distributions in the figure. To assess which distributions are different, we performed two-sample Kolmogorov-Smirnoff tests on each pair, using Bonferroni correction for multiple comparisons, resulting in an alpha level of 0.0033. Cities are significantly different from all other distributions (all $p < 0.00001$). Offices are significantly different from all others (all $p < 0.001$) except for forests ($p = 0.048$). The remaining pairs are not significantly different from one another (all $p > 0.07$).

credit. Five participants' data were excluded from the analysis due to floor performance.

The number of participants was chosen based upon a previous study with a similar stim-

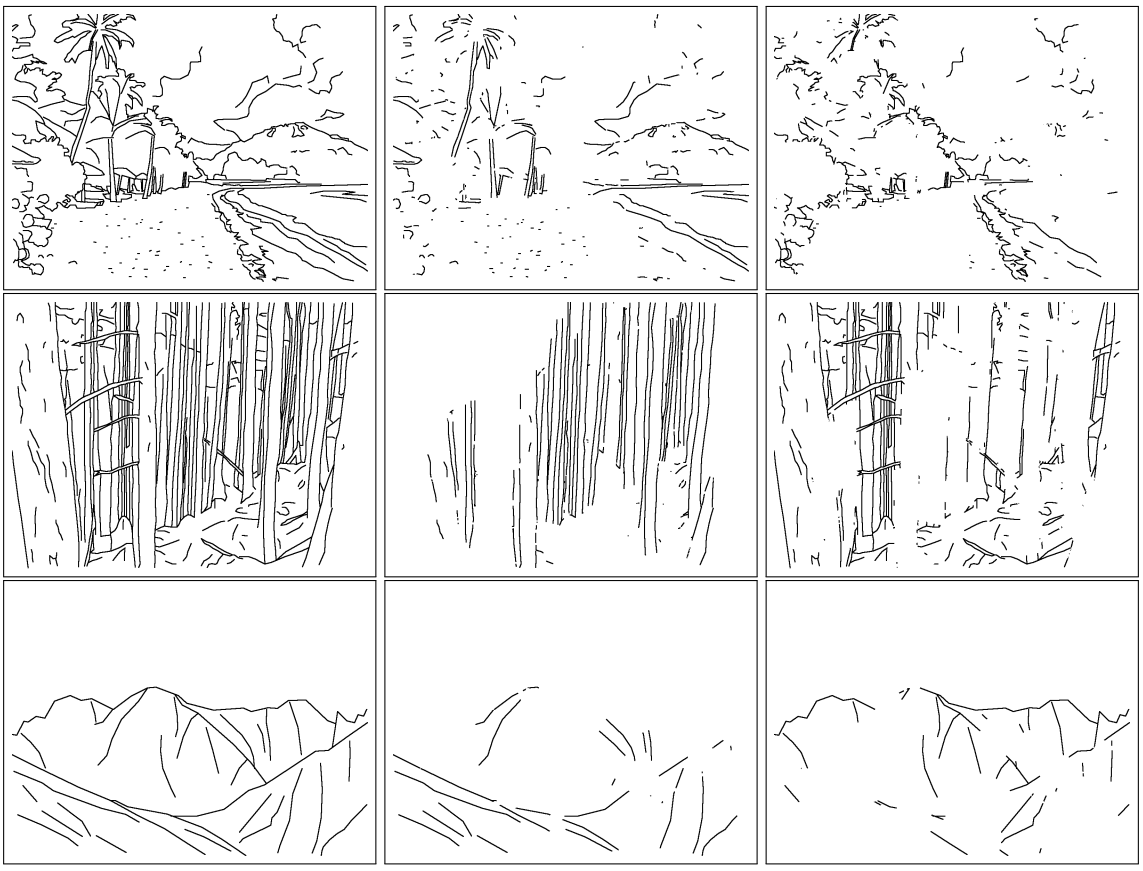


Figure 5.4: Examples for each natural scene category and condition. Rows denote category (Beaches, Forests, and Mountains), and columns denote image condition (Intact, Symmetric, Asymmetric). Note that for scenes with many contour pixels participating in strong local symmetries (e.g., the forest scene in the second row above), even the least symmetric 50% of the contour pixels can include pixels with relatively large symmetry scores.

ulus set and design [150]. All participants gave written informed consent prior to the experiment, and all procedures were approved by the University of Toronto Research Ethics Board and adhere to the tenets of the Declaration of Helsinki.

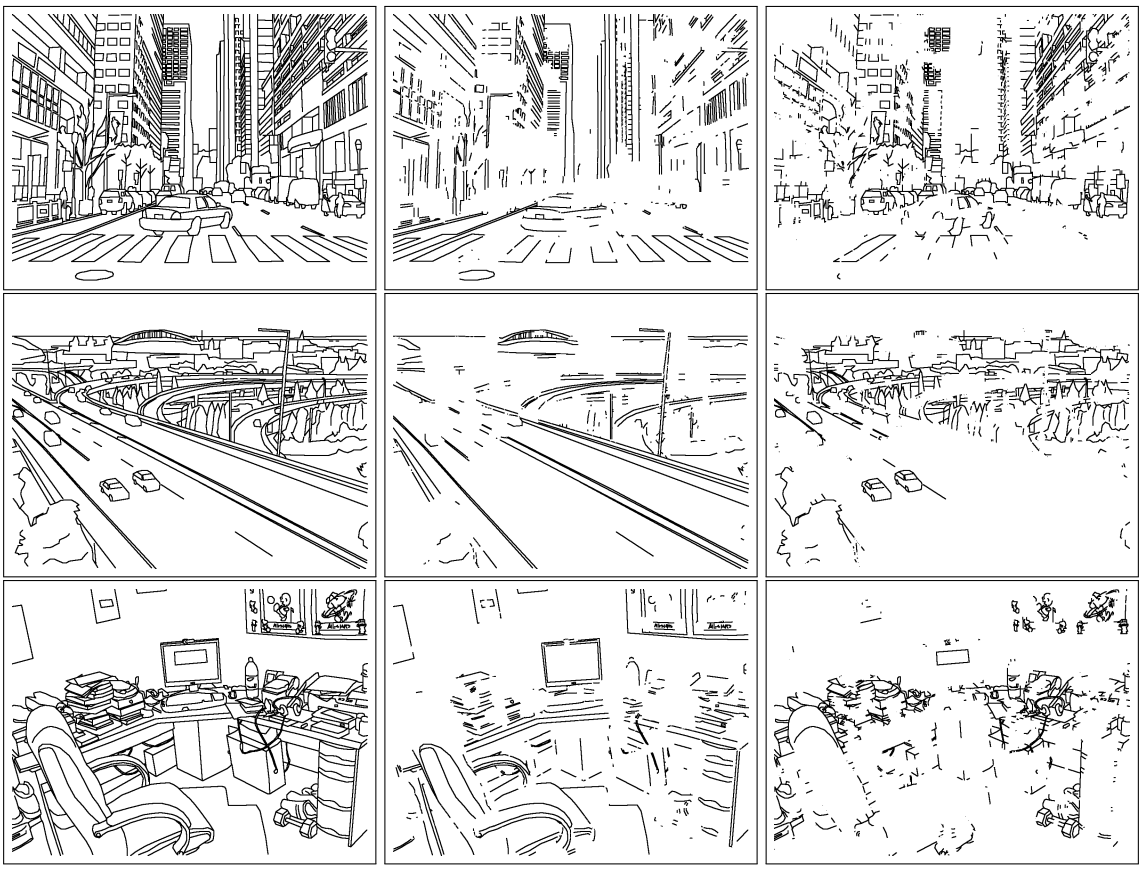


Figure 5.5: Examples for each man-made scene category and condition. Rows denote category (Cities, Highways, and Offices), and columns denote image condition (Intact, Symmetric, Asymmetric). Note that for scenes with many contour pixels participating in strong local symmetries (e.g., the forest scene in the second row above), even the least symmetric 50% of the contour pixels can include pixels with relatively large symmetry scores.

5.2.4 Design and Procedure

Participants were seated approximately 57 cm away from the monitor. The head position was not constrained. The experiment room was dark for the duration of the experiment. The experiment had three phases: Training, Ramping, and Testing.

5.2 Methods

On each trial, regardless of phase, participants were shown a line drawing of a scene. They were asked to respond with the category of the scene. The key mapping was randomized for each participant. At the start of each phase, participants were shown which key was mapped to which category. The possible keys were s, d, f, j, k, and l. The mapping was identical in the three phases but was shown at the beginning of each phase as a reminder.

Each trial started with the presentation of a scene image; the duration was dependent upon the current phase (see below). Immediately following the scene, a perceptual mask was displayed for 500 ms. The mask was a line drawing image composed of contour segments which are randomly drawn from the pool of all contours, from all scenes, from all categories. After the mask disappeared, the screen was blank until the participant responded with a key-press. After the response, the next trial began.

The training phase lasted until the participant responded correctly in 17 of the last 18 trials or 72 trials in total, whichever came first. Scene images were presented for 233 ms. On an incorrect trial, a low tone was played to provide feedback. All stimuli were intact line drawings.

The ramping phase (54 trials) started with four trials of 200 ms, followed by a linear decrease in stimulus duration from 200 ms to 33 ms. As in the training phase, participants received feedback, and only intact line drawings were shown.

The testing phase lasted for 360 trials (20 line drawings per category $\times$ 3 conditions $\times$ 6 categories). No feedback was given after the participant's response. The stimulus duration was fixed to 53 ms, which led to a performance of 70% for intact line drawings

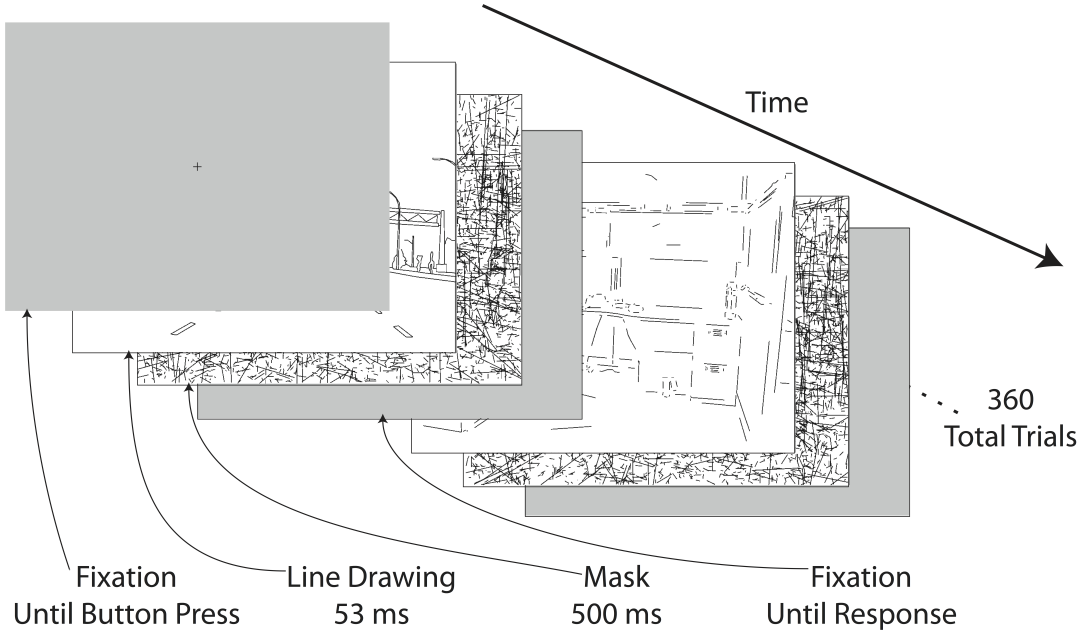


Figure 5.6: A schematic of the experiment. Stimuli (intact, symmetric, or asymmetric versions of a line drawing) were presented for 53 ms, followed by a perceptual mask for 500 ms. A blank screen was displayed until the participant responded. A total of 360 trials were presented in the testing phase.

in a pilot experiment with a different set of participants. This would result in some errors in the intact case, which allows for a comparison of the error patterns between intact and the other conditions. Each scene was only shown in one condition, and none of the test scenes were used in the previous phases so that scenes were novel on each presentation. Participants could pause between trials if they needed a break. A schematic of the test phase of the experiment can be seen in Figure 5.6.

5.3 Results

Participants most easily classified the intact line drawings (76.6 percent correct). Symmetric scenes were classified at 60.9 percent correct, and asymmetric scenes were classified at 53.3 percent correct. All conditions were well above chance performance of 16.7 percent (Figure 5.7 A). Removing any image content clearly hindered performance; performance in the symmetric condition was significantly worse than in the intact condition (paired-sample t-test, $t(20) = 13.80, p = 1.10 \times 10^{-11}$). Categorization of the intact scenes was also significantly better than asymmetric scenes (paired-sample t-test, $t(20) = 22.13, p = 1.56 \times 10^{-15}$). Crucially, the performance for symmetric scenes was significantly better than for asymmetric scenes, even though both versions of the stimuli contained the same number of contour pixels (paired-sample t-test, $t(20) = 6.21, p = 4.56 \times 10^{-6}$).

We further break down performance into the different categories (see the confusion matrices in Figure 5.8). The row labels of the confusion matrix denote the ground-truth response, and the column labels denote the response of the observers. Each cell shows the proportion of observer responses for the given ground truth category, and each row sums to 1. The diagonal elements are correct answers, and off-diagonal elements are errors. We computed correlations between the off-diagonal elements of the confusion matrices (only off-diagonal elements were used so that the overall proportion correct does not affect the correlation). The confusion matrices do not show any obvious difference in the pattern of errors in the different image conditions; all correlations between error patterns were significant: intact vs symmetric ($\rho = 0.57$, $p < 0.018$), intact vs asymmetric ($\rho = 0.74$, $p < 1.0 \times 10^{-5}$), and symmetric vs asymmetric ($\rho = 0.65$, $p < 1.0 \times 10^{-5}$).

5.3 Results

Comparing performance separately by category reveals variations in the performance for symmetric and asymmetric images. Four of the six categories showed better performance in the symmetric condition than in the asymmetric condition, leading to better average performance across all conditions. For office scenes, for instance, participants performed considerably better when seeing symmetric than asymmetric images (repeated-measures t-test, $t(20) = 4.21, p = 4.08 \times 10^{-4}$). For mountain scenes, on the other hand, performance is equal in the symmetric and asymmetric conditions ($t(20) = -0.08, p = 0.93$).

Presumably, this is due to different types of contour relationships present in the mountain scenes than in other scenes where the symmetry effect is present. For example, some objects tend to be symmetric, but very few are present in our mountain scenes. Additionally, symmetries between scene elements, such as the symmetry present between neighboring tree trunks (in a forest) or between the windows in a building (in a city) are not present in a mountain scene. Since mountain scenes lack these sorts of symmetries, removing symmetric contours leads to different distortions in mountain scenes than for other categories. Highway scenes also showed a significant performance difference ($t(20) = 4.28, p = 3.6 \times 10^{-4}$). Forest scenes showed a large performance difference between symmetric and asymmetric ($t(20) = 3.86, p = 9.78 \times 10^{-4}$). Tree trunks, with their high degree of ribbon symmetry, are prone to be distorted disproportionately when symmetric content is removed, whereas the highly irregular foliage will be present, but less recognizable, in the asymmetric images. Beach scenes also showed a modest effect ($t(20) = 2.86, p = 0.0097$), slightly smaller than that present in human-made scenes.

While there was a large effect in all human-made scenes, the direction of the effect was reversed for city scenes, relative to the direction found for all other scene categories ($t(20) = -3.31, p = 0.004$). With build environments exhibiting a high degree of structural symmetry, smaller, isolated objects, such as people and cars, frequently show comparably weaker *2D* ribbon symmetry than buildings, even though they are *3D* symmetric. As a result, such objects are almost entirely contained in the asymmetric image and maybe a strong cue to scene category. Finally, the scale of symmetry we measure may not match the scale of the symmetry that exists in a city. For example, neighboring contours may not be object boundaries but instead boundaries of regions/parts inside a single object. We will consider this in more detail when we discuss possible limitations of our symmetry scoring method.

5.4 Discussion

What drives the large difference in performance between the ribbon symmetric and ribbon asymmetric scenes? One possibility is that participants use local symmetry content as a summary statistic, either computing a single symmetry summary score for each scene or the entire distribution of symmetry content. Wilder et al. [158] demonstrated that contour cotermination at junctions had a weaker influence on scene perception than what appeared to be a longer-ranged relationship. Here we concretely measure and control parallelism in the image, and we demonstrate that local ribbon symmetry does indeed influence scene perception. Even though the symmetry measured here represents a relationship between contours, once the symmetry is measured, this information could be ignored and only the

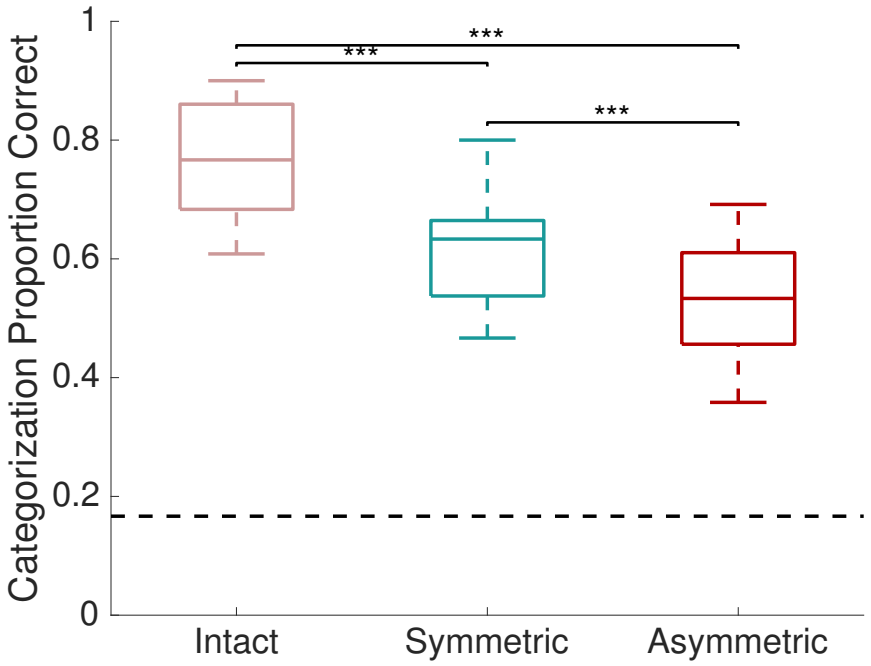


Figure 5.7: Proportion correct for each image condition. The boxplots are centered at the mean, with a line at the median. The box extends to the 25th and 75th percentiles. The lines extending from the box show the extent of all the data points. Intact categorization performance was better than either symmetric or asymmetric categorization performance ($p < 0.001$). Symmetric scenes were categorized more easily than asymmetric scenes ($p < 0.001$).

distribution of symmetry in the image used for classification. We believe this is unlikely. The distributions for a given scene category are very different in the three different conditions. Thus, participants would need to learn the distribution for each condition without prior experience with these manipulations and in the absence of feedback. Moreover, a participant's visual system would need to know which condition they are seeing in order to accurately use this information. If this were the case, we would expect different error

5.4 Discussion

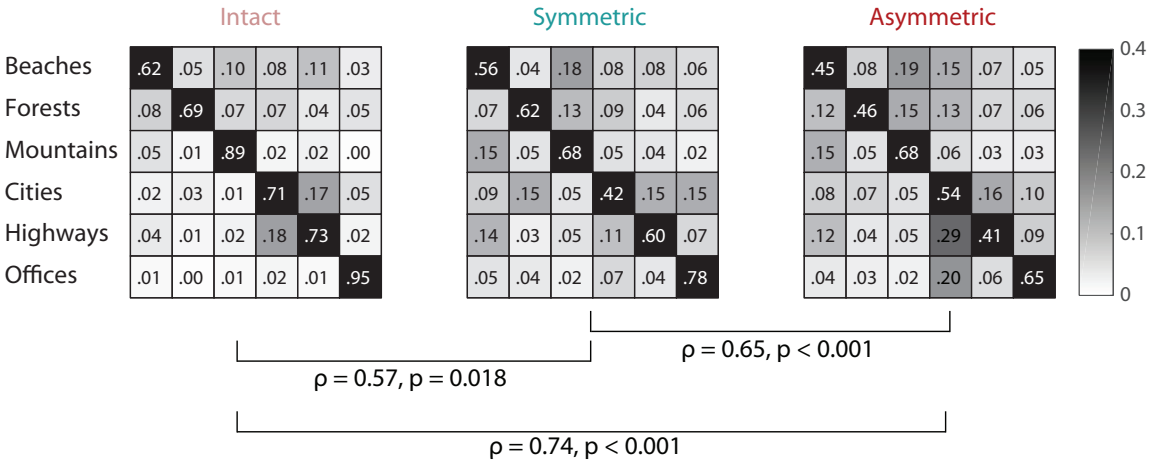


Figure 5.8: Confusion matrices for the different conditions. Rows are the true category labels, and the columns are the subject responses. Correct answers lie on the diagonal, so a strong diagonal represents good performance.

patterns in the asymmetric condition (without strong symmetry content) than in either the intact or symmetric conditions (with strong symmetry content), but the confusion matrices show no obvious difference in error patterns (Figure 5.8). We hypothesize that the visual system uses symmetry to jump-start the grouping of image information into meaningful units. The performance was lower in the asymmetric condition because the process could not be jump-started to the same extent, resulting in a poorer grouping.

Walther et al. [151] suggested that longer contours are more useful for scene classification. When selecting the most/least symmetric contour pixels we did not control the length of contiguous sets of contour pixels. Thus, the average length of contiguous contour segments is not necessarily equal in the two half-images. While this could play a role in performance, we argue that it is not responsible for our large performance difference.

In order to estimate the length of each segment, we selected a set of connected black

pixels and counted the number of pixels in that set. Figure 5.9 shows histograms of contour length for the symmetric (turquoise) and asymmetric (red) images combined across all categories. The average length is shorter in asymmetric than symmetric images. Note, however, that the variance in contour length is larger for asymmetric images, as they also tend to contain many of the longest contours. If, as Walther et al. [151] showed, longest contours convey the most information about scene content, the asymmetric condition should benefit, as these most informative contours live in the asymmetric images. Thus, contour length does not appear to drive the behavioral difference between symmetric and asymmetric images in our data.

Additionally, Walther et al. [151] only found a performance difference between long and short contours when 75% of the pixels were removed. When they removed 50% of the pixels, on the same line drawing used here, performance for the long and short contour versions was statistically equal. Therefore, we should not expect a performance difference here based on line length alone.

Previous work, where portions of contours were deleted, has argued that the important contour pieces were junctions [11], or the straight portions between junctions [73]. Previous work in [158] found that for scenes, middle segments between junctions were more useful for determining scene category than were junctions. The current study finds that scene perception is aided by segments participating in a symmetry relationship with another segment rather than by middle segments in general.

How does ribbon symmetry facilitate grouping image content into meaningful surface or object parts? Most objects are not mirror-symmetric in the image plane, and many ob-

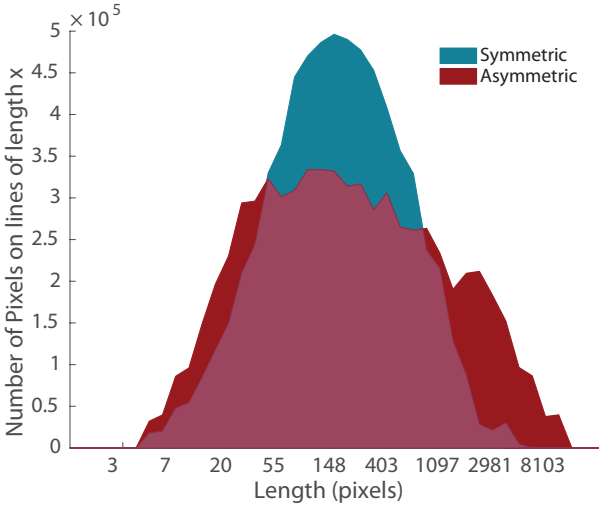


Figure 5.9: Histogram showing the length of contours in the symmetric and asymmetric images for all line-drawings in the data-set. Note that the x-axis is on a log scale.

jects that are locally symmetric will have low symmetry scores assigned to portions of their boundary contours. As a consequence, pixels from a single object can be assigned to different half-images, splitting the object into pieces. Additionally, some objects entirely fall into the asymmetric image. Many objects that are 3D mirror-symmetric in the real world are not 2D mirror-symmetric when projected onto the image plane. However, local symmetry between the boundaries of a part of an object can persist, and we conjecture that this property is used by the visual system to help group these image elements into surfaces and objects. Symmetry has been the basis of many prominent parts-based representations [11, 64], and symmetry has been shown to be involved in image segmentation [93]. Symmetry may assist in perceptual grouping, as ribbon symmetry has been shown to attract attention during scene viewing [35]. Our data suggest that when ribbon symmetric contours are removed, the image is much harder to understand. We hypothesize this is

because the observers are unable to segment the scene into meaningful objects and parts. In the asymmetric image, the symmetric contour portions required for grouping contours into objects and object parts are missing. Thus, scene categorization performance will deteriorate, if it relies on object classification.

There are limitations to our results. Our symmetry score measures the extent to which there is a constant distance between a pair of contours (i.e., ribbon symmetry). Consider a rectangle; the contours along the main axis score highly, but due to the nature of the medial axis, a lower score is assigned to the contours on the short sides. Additionally, our score is based upon the medial axis, which captures the relationship between contours that bound the same region in the scene, and will not capture the symmetry relationships between contours separated by intervening contours. Comparing all possible symmetry relationships between all contours in the image is intractable, but some consideration of longer-range symmetries is worth pursuing.

We are not claiming that the visual system relies only on symmetry when rapidly classifying a scene. Here we have only considered local ribbon symmetry in order to understand its power in isolation. Other features, such as contour junctions, may also contribute to scene categorization [150]. Combining these features with local symmetry and other longer-range symmetry relationships could provide a more complete explanation of human scene categorization.

Asymmetric images of cities were more easily categorized, which is the reverse effect of the other categories. This demonstrates the aforementioned limitations and makes apparent another. The city scenes in our data-set contain many windows, and the two longer

edges of a window will appear in the symmetric image if and only if they are sufficiently elongated; the opposing parallel sides are missed. Also, the scale of symmetry in our current measure may not be optimal for a city scene. While the symmetry of a single-window may be important, the symmetry between the sides of a single building is also important, and our method does not look at symmetry relationships at larger spatial scales. In other categories, such as forests, long-range symmetry may be less important, since the symmetric pairs tend to be the boundaries of single objects (i.e., tree trunks), which is one reason why our manipulation resulted in much better performance for symmetric forest scenes than asymmetric ones.

The non-accidental relation of symmetry was noted by the Gestalt psychologists almost a century ago [77, 154, 79] and reflects the ubiquity of symmetry in both our natural and human-made world. Given this regularity, it was not surprising that symmetry became a powerful basis for parts-based object representations in both human and computer vision [18, 1, 106, 11]. Symmetry has been carefully studied for object recognition in images containing single objects. Less attention has been paid to the role of symmetry in the perception of complex scenes which contain many objects and surfaces. The complexity of a cluttered scene has encouraged approaches that are global in nature, focusing on global scenes statistics which, in turn, avoids the challenging problem of perceptual grouping. This study represents the first attempt to study the role of a quantitative measure of local ribbon symmetry in the categorization of line drawings of complex, natural scenes. We focused on the non-accidental property of symmetry, arguably the most powerful form of perceptual grouping for relating elements at a distance, and we introduced a novel scene statistic based on the medial axis. We demonstrated that in two subsets of

a stimulus, each with the exact same number of black pixels, the subset with the stronger symmetry leads to significantly better scene perception.

The obvious question raised by our findings is "why does symmetry offer an advantage to scene perception?" Our hypothesis is that the importance of correct contour grouping is even more critical in a cluttered scene, in which any given contour may be proximal to many contours belonging to other objects. Under such conditions, where proximity leads to highly ambiguous groupings, adding symmetry cues can reduce ambiguity and lead to a better grouping of contours into surfaces that comprise object parts and, in turn, the objects that make up a scene.

Our work shows that local ribbon symmetry is a key feature that allows for the rapid analysis of complex real-world scenes. This finding lends further support to previous work on the importance of local details of the structure of a scene for rapid scene perception [151, 150, 28]. Incorporating principles of the perceptual organization originally proposed by the Gestalt psychologists in a computationally rigorous way is a promising avenue toward a more complete understanding of the computational processes that make vision appear so natural and effortless.

6

Medial Axis Based Salience Measures for Scene Categorization

The contents of this chapter are largely based on the articles "Medial Axis Based Contour Salience for Scene Categorization" [119] and "Gestalt-based Contour Weights Improve Scene Categorization by CNNs" [120] which grew out of a collaboration with colleagues in the human and computer vision groups at the University of Toronto.

The computer vision community has witnessed recent advances in scene categorization from images, with the state of the art systems now achieving impressive recognition rates on challenging benchmarks. Such systems have been trained on photographs which include color, texture and shading cues. The geometry of shapes and surfaces, as conveyed

by scene contours, is not explicitly considered for this task. Remarkably, humans can accurately recognize natural scenes from line drawings, which consist solely of contour-based shape cues. Here we report the first computer vision study on scene categorization of line drawings derived from popular databases including an artist scene database, MIT67, and Places365. Specifically, we use off-the-shelf pre-trained Convolutional Neural Networks (CNNs) to perform scene classification given only contour information as input and find performance levels well above chance. We also show that medial-axis based contour saliency methods, such as those introduced in Chapter 5 of this thesis, can be used to select more informative subsets of contour pixels and that the variation in CNN classification performance on various choices for these subsets is qualitatively similar to that observed in human performance. Moreover, when the salience measures are used to weight the contours, we find that these weights boost our CNN performance above that for unweighted contour input. That is, the medial axis based salience weights appear to add useful information that is not available when CNNs are trained to use contours alone.

6.1 Introduction

Both biological and artificial vision systems are confronted with a potentially highly complex assortment of visual features in real-world scenarios. The features need to be sorted and grouped appropriately in order to support high-level visual reasoning, including the recognition or categorization of objects or entire scenes. In fact, scene categorization cannot be easily disentangled from the recognition of objects, since scene classes are often defined by a collection of objects in context. A beach scene, for example, would typically

contain umbrellas, beach chairs and people in bathing suits, all of whom are situated next to a body of water. A street scene might have roads with cars, cyclists, and pedestrians as well as buildings along the edge. How might computer vision systems tackle this problem of organizing visual features to support scene categorization?

In human vision, perceptual organization is thought to be affected by a set of heuristic grouping rules originating from Gestalt psychology [77]. Such rules posit that visual elements ought to be grouped together if they are, for instance, similar in appearance, in close proximity, or if they are symmetric or parallel to each other. Developed on an ad-hoc, heuristic basis originally, these rules have been validated empirically, even though their precise neural mechanisms remain elusive. Grouping cues, such as those based on symmetry, are thought to aid in high-level visual tasks such as object detection because symmetric contours are more likely to be caused by the projection of a symmetric object than to occur accidentally. In the categorization of complex real-world scenes by human observers, local contour symmetry does indeed provide a perceptual advantage [159], but the connection to the recognition of individual objects is not as straightforward as it may appear.

In computer vision, symmetry, proximity, good continuation, contour closure, and other cues have been used as we showed in Chapter 5, for image segmentation, curve inference, object recognition, object manipulation, and other tasks [96, 11, 45, 123]. Instantiations of such organizational principles have found their way into many computer vision algorithms and have been the subject of regular workshops on the perceptual organization in artificial vision systems. However, perceptually motivated salience measures

6.1 Introduction

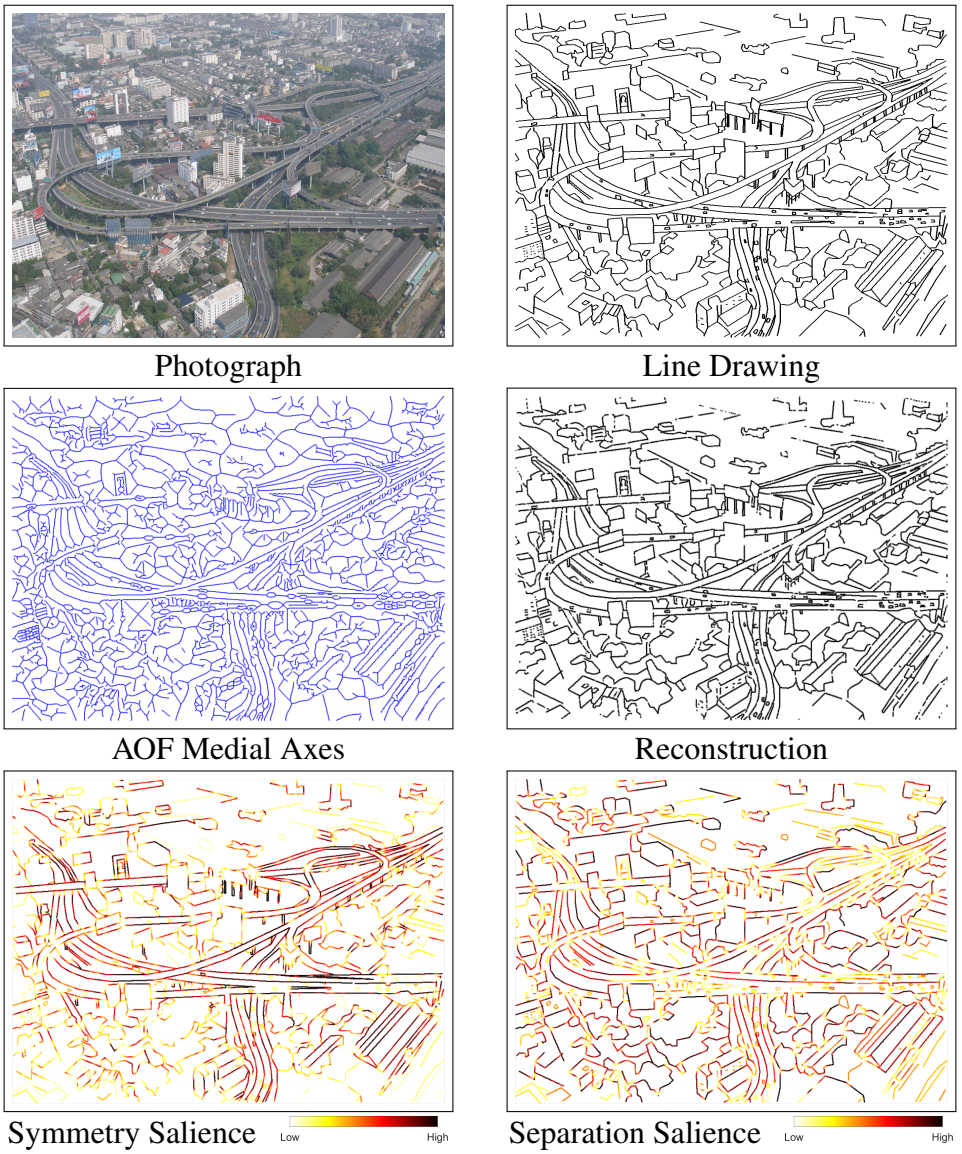


Figure 6.1: (Best viewed by zooming in on the PDF.) An illustration of our approach on an example from a database of line drawings by artists of photographs of natural scenes. The middle right panel shows the reconstruction of the artist-generated line drawing from the AOF medial axes. The bottom panels present a hot colormap visualization of two of our medial axis based contour salience measures.

to facilitate scene categorization have received little attention thus far. This may be a result of the ability of CNN-based systems to accomplish scene categorization on challenging databases, in the presence of sufficient training data, directly from pixel intensity and color in photographs [128, 141, 66, 165]. CNNs begin by extracting simple features, including oriented edges, which are then successively combined into more and more complex features in a succession of convolution, nonlinear activation, and pooling operations. The final levels of CNNs are typically fully connected, which enables learning of object or scene categories [139, 4, 59, 115]. Unfortunately, present CNN architectures do not explicitly allow for properties of object shape to be represented explicitly. This limitation has been recognized and is the subject of some promising new work in the field [57, 56]. Human observers, in contrast, recognize an object's shape as an inextricable aspect of its properties, along with its category or identity [72].

Comparisons between CNNs and human and monkey neurophysiology appear to indicate that CNNs replicate the entire visual hierarchy [62, 22]. Does this mean that the problem of perceptual organization is now irrelevant for computer vision? In the present chapter, we argue that this is not the case. Rather, we show that CNN-based scene categorization systems, just like human observers, can benefit from explicitly computed contour measures derived from Gestalt grouping cues. We here demonstrate the computation of these measures as well as their power to aid in the categorization of complex real-world scenes.

To effect our study, with its focus on the geometry of scene contours, as in other chapters in this thesis we use the medial axis transform (MAT) as a representation. We apply

the same algorithm for computing the medial axis to analyze line drawings of scenes of increasing complexity (see Section 2.1), using average outward flux of the gradient of the Euclidean distance function through shrinking circular disks [38]. With its explicit representation of the regions between scene contours, the medial axis allows us to directly capture salience measures related to local contour separation and local contour symmetry. We introduce two novel measures of local symmetry using ratios of length functions derived from the medial axis radius along with skeletal segments. Distinct from the approach in Chapter 5, these new measures have clearer geometric interpretations and have the further advantage that they are essentially parameter-free. As ratios of commensurate quantities, these are unitless measures, which are therefore invariant to image re-sizing. We also introduce a measure of local contour separation. We describe methods of computing our perceptually motivated salience measures from line drawings of photographs of complex real-world scenes, covering databases of increasing complexity. Figure 6.1 presents an illustrative example of a photograph from an Artist Scenes database, along with two of our medial axis based contour saliency maps. Observe how the ribbon symmetry based measure highlights the boundaries of highways similar to Figure 5.5, where parallel contours in scenes are shown to facilitate categorization in scenes by humans. Our experiments will show that scene contours weighted by these measures can boost CNN-based scene categorization accuracy, despite the absence of color, texture and shading cues. Our work indicates that measures of contour grouping, that are simply functions of the contours themselves, are beneficial for scene categorization by computers, yet that they are not automatically extracted by state-of-the-art CNN-based scene recognition systems. The critical remaining question is whether this omission is due to the CNN archi-

tecture being unable to model these weights or whether this has to do with the (relatively standard) training regime. We leave this for further study.

6.2 Medial Axis Based Contour Saliency

Owing to the continuous mapping between the medial axis and scene contours, the medial axis provides a convenient representation for designing and computing Gestalt contour salience measures based on local contour separation and local symmetry. A measure to reflect local contour separation can be designed using the radius function along the medial axis since this gives the distance to the two nearest scene contours on either side. Local parallelism between scene contours, or ribbon symmetry, can also be directly captured by examining the degree to which the radius function along the medial axis between them remains locally constant. Finally, if the taper is to be allowed between contours, as in the case of a set of railway tracks extending to the horizon under perspective projection, one can examine the degree to which the first derivative of the radius function is constant along a skeletal segment. We introduce novel measures to capture local separation, ribbon symmetry and taper, based on these ideas.

In the following we shall let p be a parameter that runs along a medial axis segment, $\mathbf{C}(p) = (x(p), y(p))$ be the coordinates of points along that segment, and $r(p)$ be the medial axis radius at each point. We shall consider the interval $p \in [\alpha, \beta]$ for a particular medial segment. The arc length of that segment is given by

$$L = \int_{\alpha}^{\beta} ||\frac{\partial \mathbf{C}}{\partial p}||dp = \int_{\alpha}^{\beta} (x_p^2 + y_p^2)^{\frac{1}{2}} dp. \tag{6.1}$$

6.2.1 Separation Salience

We now introduce a salience measure based on the local separation between two scene contours associated with the same medial axis segment. Consider the interval $p \in [\alpha, \beta]$. With $r(p) > 1$ in pixel units (because two scene contours cannot touch) we introduce the following contour separation based salience measure:

$$S_{Separation} = 1 - \Big(\int_{\alpha}^{\beta} \frac{1}{r(p)} dp \Big) / (\beta - \alpha). \tag{6.2}$$

This quantity falls in the interval $[0, 1]$. The measure increases with increasing spatial separation between the two contours. In other words, scene contours that exhibit further (local) separation are more salient by this measure.

6.2.2 Ribbon Symmetry Salience

Now consider the curve $\Psi = (x(p), y(p), r(p))$. Similar to Equation 6.1, the arc length of Ψ is computed as:

$$L_{\Psi} = \int_{\alpha}^{\beta} || \frac{\partial \Psi}{\partial p} || dp = \int_{\alpha}^{\beta} (x_p^2 + y_p^2 + r_p^2)^{\frac{1}{2}} dp. \tag{6.3}$$

When two scene contours are close to being parallel locally, $r(p)$ will vary slowly along the medial segment. This motivates the following ribbon symmetry salience measure:

$$S_{Ribbon} = \frac{L}{L_{\Psi}} = \frac{\int_{\alpha}^{\beta} (x_p^2 + y_p^2)^{\frac{1}{2}} dp}{\int_{\alpha}^{\beta} (x_p^2 + y_p^2 + r_p^2)^{\frac{1}{2}} dp}. \tag{6.4}$$

This quantity also falls in the interval $[0, 1]$ and is invariant to image scaling since the integral involves a ratio of unitless quantities. The measure is designed to increase as the scene contours on either side become more parallel, such as the two sides of a ribbon.

6.2.3 Taper Symmetry Salience

A notion that is closely related to that of ribbon symmetry is taper symmetry; two scene contours are taper symmetric when the medial axis between them has a radius function that is changing at a constant rate, such as the edges of two parallel contours in 3D when viewed in perspective. To capture this notion of symmetry, we introduce a slight variation where we consider a type of arc-length of a curve $\Psi' = (x(p), y(p), \frac{dr(p)}{dp})$. Specifically, we introduce the following taper symmetry salience measure:

$$S_{Taper} = \frac{L}{L_{\Psi'}} = \frac{\int_{\alpha}^{\beta}(x_p^2 + y_p^2)^{\frac{1}{2}}dp}{\int_{\alpha}^{\beta}(x_p^2 + y_p^2 + (rr_{pp})^2)^{\frac{1}{2}}dp}. \tag{6.5}$$

The bottom integral is not exactly an arc-length, due to the multiplication of r_{pp} by the factor r. This modification is necessary to make the overall ratio unitless. This quantity also falls in the interval $[0, 1]$ and is invariant to image scaling. The measure is designed to increase as the scene contours on either side become more taper symmetric, as in the shape of a funnel, or the sides of a railway track.

To gain an intuition behind these perceptually driven contour salience measures, we provide three illustrative examples in Fig. 6.2. The measures are not computed point-wise, but rather for a small interval $[\alpha, \beta]$ centered at each medial axis point (see Section 6.3.3 for details). When the contours are parallel, all three measures are constant along the

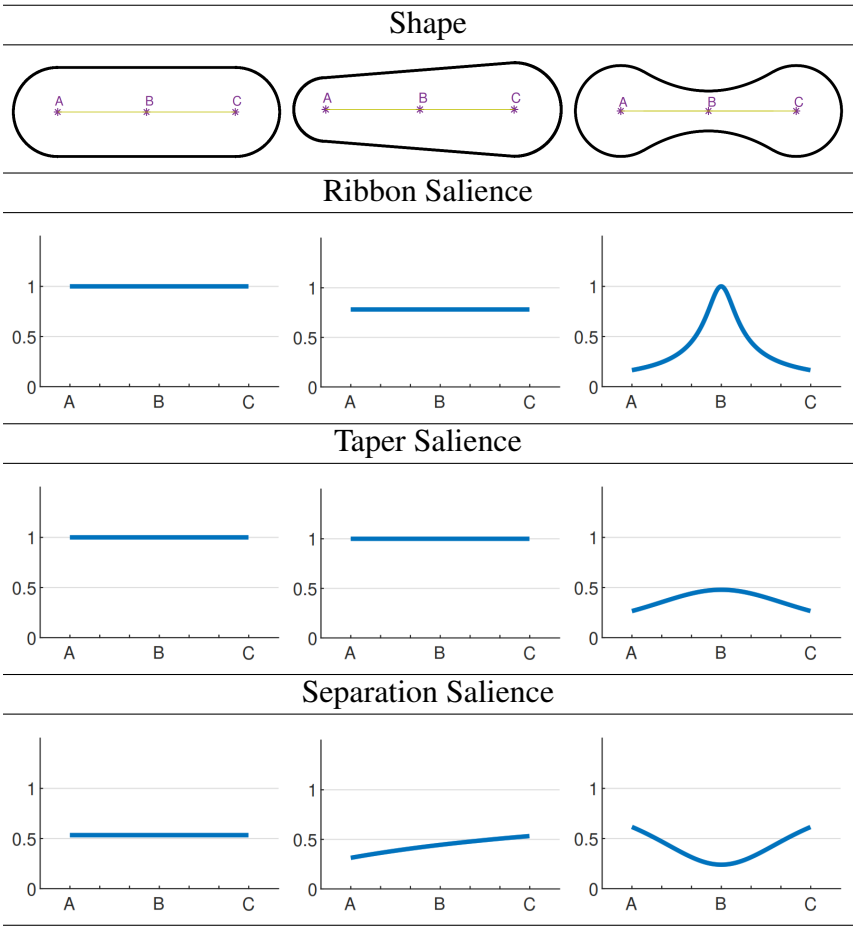


Figure 6.2: An illustration of ribbon symmetry salience, taper symmetry salience and contour separation salience for three different contour configurations. See text for a discussion. These measures are all invariant to 2D similarity transforms of the input contours

medial axis (left column). The middle figure has high taper symmetry but lower ribbon symmetry, with contour separation salience increasing from left to right. Finally, for the dumbbell shape, all three measures vary (third column).

6.3 Experiments and Results

6.3.1 Artist Generated Line Drawings

Artist Scenes Database: Color photographs of six categories of natural scenes (beaches, city streets, forests, highways, mountains, and offices) were downloaded from the internet, and those rated as the best exemplars of their respective categories by workers on Amazon Mechanical Turk were selected. Line drawings of these photographs were generated by trained artists at the Lotus Hill Research Institute [151]. Artists traced the most important and salient lines in the photographs on a graphics tablet using a custom graphical user interface. Contours were saved as successions of anchor points. For the experiments in the present chapter, line drawings were rendered by connecting anchor points with straight black lines on a white background at a resolution of 1024×768 pixels. The resulting database had 475 line drawings in total with 79-80 exemplars from each of 6 categories: beaches, mountains, forests, highway scenes, city scenes, and office scenes.

6.3.2 Machine Generated Line Drawings

MIT67/Places365 Given the limited number of scene categories in the Artist Scenes database, particularly for computer vision studies, we worked to extend our results to the two popular but much larger scene databases of photographs - MIT67 [114] (6700 images, 67 categories) and Places365 [165] (1.8 million images, 365 categories). Producing artist-generated line drawings on databases of this size was not feasible, so instead, we came up with ways to generate such line drawings by computer.

6.3 Experiments and Results

Initially, in our first set of experiments, we fine-tuned the output of the Dollar edge detector [39], using the publicly available structured edge detection toolbox. From the edge map and its associated edge strength, we produced a binarized version, using per image adaptive thresholding. The binarized edge map was then processed to obtain contour fragments of width 1 pixel. Each contour fragment was then spatially smoothed by convolution of the coordinates of points along with it, using a Gaussian with $\sigma = 1$, to mitigate discretization artifacts. The same parameters were used to produce all the MIT67 and Places365 line drawings. We confirmed that on the artist's line drawing database 90% of the machine-generated contour pixels were in common with the artist's line drawings. Figure 6.5 shows several typical machine-generated line drawings from the MIT67 and Places365 databases but weighted by our perceptual salience measures. CNN based scene categorization results using Dollar's edge detector have been reported in [119].

Later in the lifetime of this project, to obtain more accurate results, we migrated from Dollar's edge detection algorithm to another framework. This time, we modified the output of the Logical/Linear edge detector [68], using their publicly available open-source implementation. This approach is devised to recover image curves while preserving singularities and junctions. We briefly review the three kinds of image curves modeled in [68]. Figure 6.4 presents a comparison of machine-generated and artist-generated line drawings for an office scene from the Artist Scenes database.

Consider an image $I : \mathbb{R}^2 \rightarrow \mathbb{R}^+$, with $P = [\alpha, \beta]$ and let $\mathbf{C} : p \in P \rightarrow \mathbb{R}^2$ represent a smooth curve parameterized by arc length (see Figure 6.3). The normal cross section

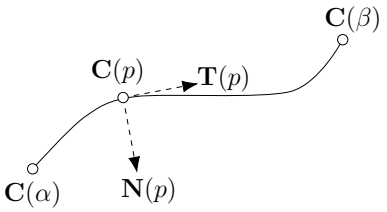


Figure 6.3: An image curve shown as $C : p \in P \rightarrow \mathbb{R}^2$ with unit tangent vector $\mathbf{T}(p)$, and unit normal vector $\mathbf{N}(p)$.

$\mathbf{N}_p(t)$ at the curve point $\mathbf{C}(p)$ is given by:

$$\mathbf{N}_p(t) = I(\mathbf{C}(p) + t\mathbf{N}(p))), \quad p \in P, \quad t \in \mathbb{R}. \tag{6.6}$$

Using local structural conditions in the directions tangential and normal to the curve, the following three image curve categories are suggested in [68]:

1. $\mathbf{C}$ is an *Edge* iff $\mathbf{C}$ is an image curve such that the following condition holds for all $p \in P$:

$$\lim_{t \to 0^-} \mathbf{N}_p(t) > \lim_{t \to 0^+} \mathbf{N}_p(t)$$

2. $\mathbf{C}$ is a *Positive Constrast Line* iff $\mathbf{C}$ is an image curve such that the following condition holds for all $p \in P$:

$$\lim_{t \to 0^-} \mathbf{N'}_p(t) > 0 \text{ and } \lim_{t \to 0^+} \mathbf{N'}_p(t) < 0$$

3. $\mathbf{C}$ is a *Negative Constrast Line* iff $\mathbf{C}$ is an image curve such that the following

condition holds for all $p \in P$:

$$\lim_{t \to 0^-} \mathbf{N'}_p(t) < 0 \text{ and } \lim_{t \to 0^+} \mathbf{N'}_p(t) > 0$$

In [68] operators are designed to respond when any of the above conditions are met locally in an image, and if so, either an edge or a line is reported. In our experiments we focused on the case of edge points; from the output edge map and its associated edge strength and edge directions, we produced a binarized version. Each binarized edge map was processed and traced to obtain contour fragments having a width of 1 pixel.

6.3.3 Computing Contour Salience

Computing contour salience for each line drawing required a number of steps. First, each connected region between scene contours was extracted. Second, we computed an AOF map for each of these connected components, as explained in Chapter 2.1. For this we used a disk of radius 1 pixel, with 60 discrete sample points on it, to estimate the AOF integral. We used a threshold of $\tau = 0.25$ on the AOF map, which corresponds to an object angle $\theta \approx 23$ degrees, to extract skeletal points. A typical example appears in Figure 6.1 (middle left). The resulting AOF skeleton was then partitioned into medial curves between branch points or between a branch point and an endpoint. We then computed a discrete version of each of the three salience measures in Section 6.2, within an interval $[\alpha, \beta]$ of length $2K + 1$, centered at each medial axis point, with $K = 5$ pixels. Each scene contour point was then assigned the maximum of the two salience values at the closest points on the medial curves on either side of it, as illustrated in Figure 6.1 (bottom left and bottom

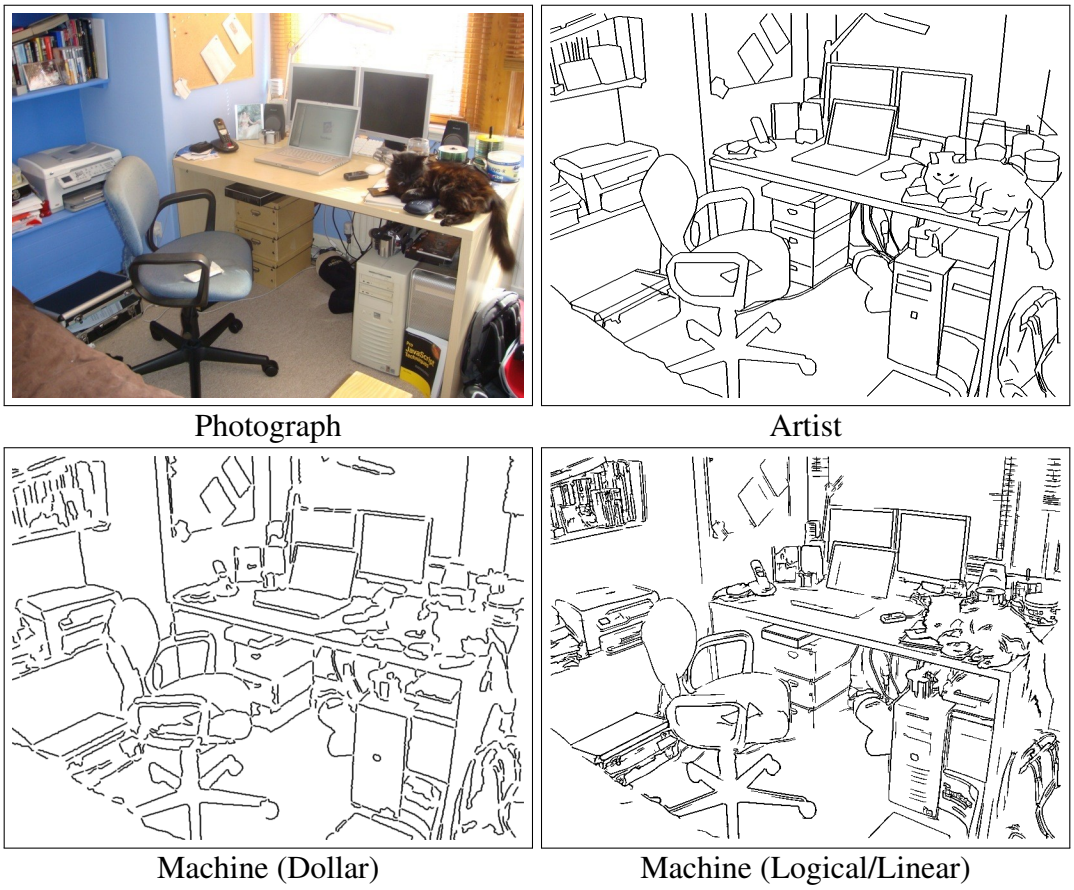


Figure 6.4: (Best viewed by zooming in on the PDF.) A comparison between machine-generated line drawings (Dollar [39] and Logical/Linear [68]) and one drawn by an artist, for an office scene from the Artist Scenes database.

right).

6.3.4 Experiments on 50-50 Splits of Contour Scenes

Our first set of experiments is motivated by our earlier work that shows that human observers benefit from contour symmetry in scene recognition from contours [159], which

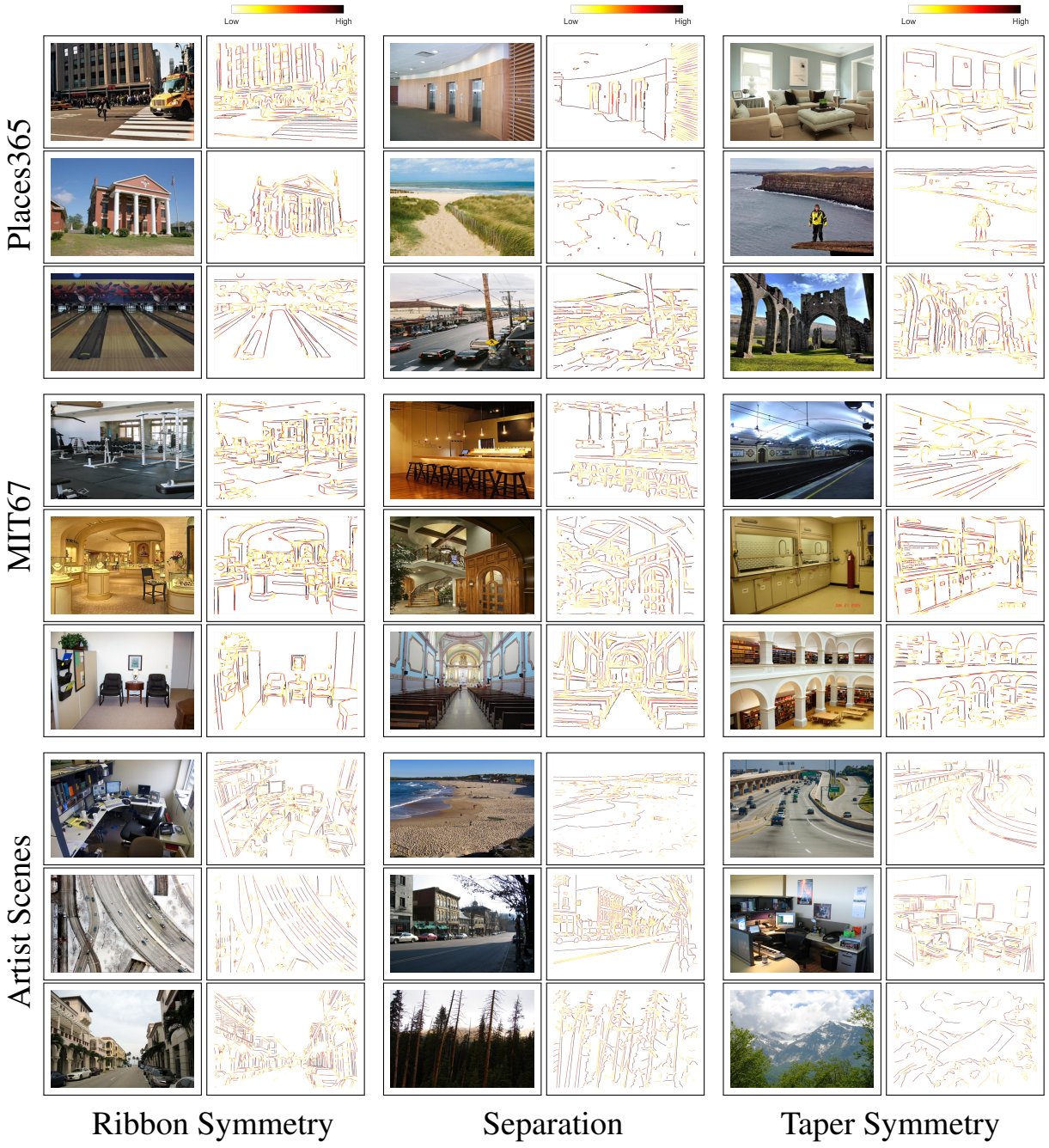


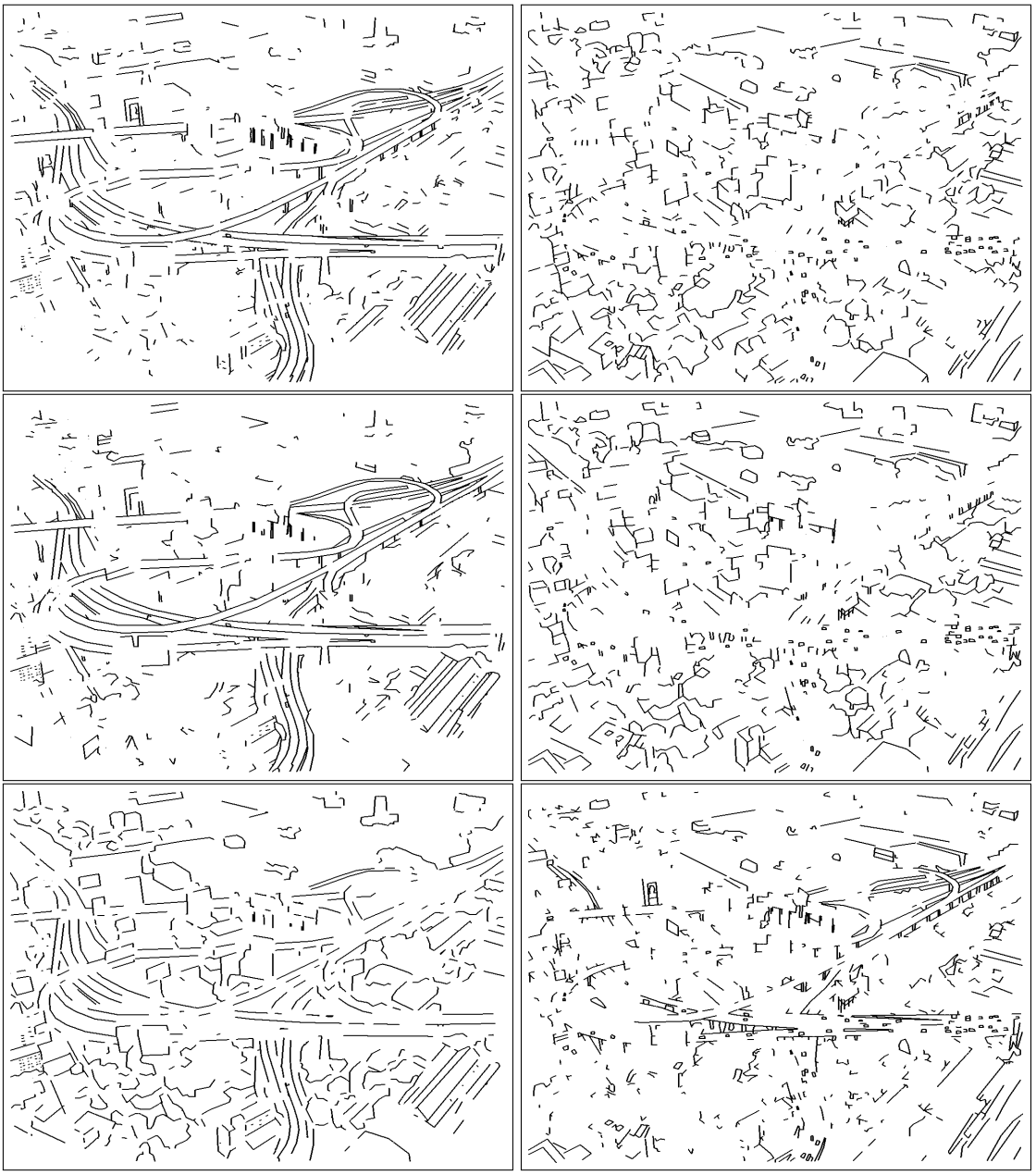
Figure 6.5: (Best viewed by zooming in on the PDF.) Examples of original photographs and the corresponding ribbon symmetry salience weighted, separation salience weighted and taper symmetry salience weighted scene contours, using a hot colormap to show increasing values. Whereas the Artist Scenes line drawings were produced by artists, these MIT67 and Places365 line drawings were machine-generated using Dollar's edge detector[39].

was presented in Chapter 5. Our goal is to examine whether a CNN-based system also benefits from such perceptually motivated cues. Accordingly, we created splits of the top 50% and the bottom 50% of the contour pixels in each image of the Artist Scenes and MIT67 data sets, using the three salience measures, ribbon symmetry, taper symmetry, and local contour separation. An example of the original intact line drawing and each of the three sets of splits is shown in Figure 6.6, for the highway scene from the Artist Scenes dataset shown in Figure 6.1.

On the Artist Scenes dataset, human observers were tasked with determining to which of six scene categories an exemplar belonged. The input was either the artist-generated line drawing or the top or the bottom half of a split by one of the salience measures. Images were presented for only 58 ms and were followed by a perceptual mask, making the task difficult for observers, who would otherwise perform near 100% correct. The results with these short image presentation durations, shown in Figure 6.7 (top), demonstrate that human performance is consistently better with the top (more salient) half of each split than the bottom one, for each salience measure. The human performance is slightly boosted for all conditions in the separation splits, for which a different subject pool was used.

Carrying out CNN-based recognition on the Artist Scenes and MIT67 line drawing datasets presents the challenge that they are too small to train a large model, such as VGG-16, from scratch. To the best of our knowledge, no CNN-based scene categorization work has so far focused on line drawings of natural images. We, therefore, use CNNs that are pre-trained on RGB photographs for our experiments.

For our experiments on the Artist and MIT67 datasets (using Dollar's edge detector[39]



Top 50% Bottom 50%

Figure 6.6: We consider the same highway scene as in Figure 6.1 (top left) and create splits of the artist generated line drawings, each of which contains 50% of the original pixels, based on ribbon symmetry (top row), taper symmetry (middle row) and local contour separation (bottom row) based salience measures. In each row the more salient half of the pixels is on the left.

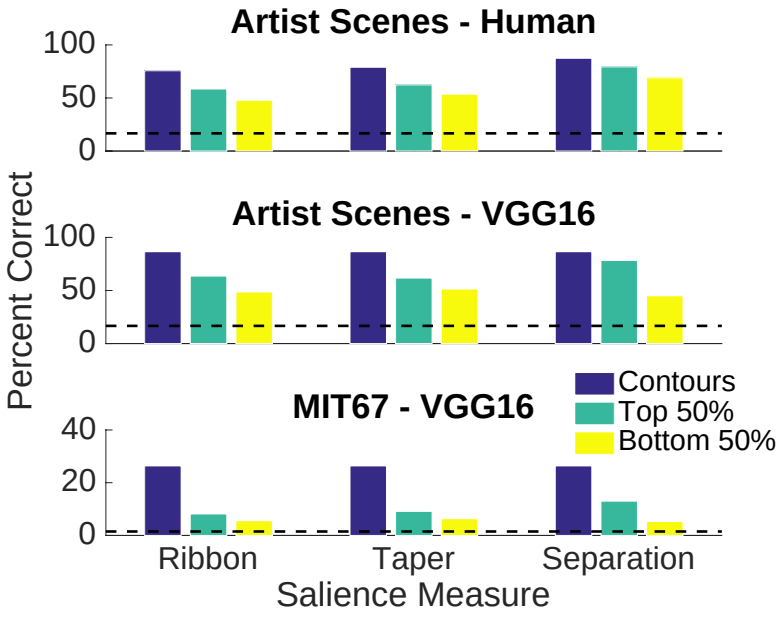


Figure 6.7: A comparison of human scene categorization performance (top row) with CNN performance (middle and bottom rows). As with the human observer data, CNNs perform better on the top 50% half of each split according to each salience measure, that the bottom 50% half. In each plot chance level performance (1/6 for Artist Scenes and 1/67 for MIT67) is shown with a dashed line.

for machine-generated line drawings), we use the VGG16 convolutional layer network architecture [137] with weights pre-trained on ImageNet. The last three layers of the VGG16 network used for fine-tuning are replaced with a fully connected layer, a softmax layer and a classification layer, where the output label is one of the categories in each of our datasets. The images are processed by this network and the final classification layer produces an output vector in which the top-scoring index is selected as the prediction output. For the Places365 dataset, which contains 1.8 million images, we used Resnet50 [63] with its weights obtained by training on ImageNet, but rather than fine-tune the network, we used

CNN	Human
Ribbon Sym vs Asymm $t(4) = 26.12$ $p = 1.3E{-}5$	Ribbon Sym vs Asymm $t(25) = 7.86$ $p = 3.2E{-}8$
Taper Sym vs Asym $t(4) = 12.39$ $p = 2.4E{-}4$	Taper Sym vs Asym $t(25) = 6.46$ $p = 9.2E{-}7$
Separation Far vs Near $t(4) = 100.64$ $p = 5.85E{-}8$	Separation Far vs Near $t(5) = 5.2$ $p = 3.0E{-}3$

Table 6.1: T-tests results for CNN and human categorization experiments.

the final fully connected layer output as a feature vector input to an SVM classifier. For all experiments on the Artist Scenes, we use 5-fold cross-validation. Top-1 classification accuracy is given, as a mean over the 5 folds, in Figure 6.7 (middle). The CNN-based system mimics the trend we saw in human observers, namely that performance is consistently better for the top 50% of each of the three splits. We interpret this as evidence that all three Gestalt motivated salience measures are beneficial for scene categorization in both computer and human vision.

For MIT67 we use the provided training/test splits and present the average results over 5 trials. The CNN-based categorization results are shown in Figure 6.7 (bottom row). It is striking that even for this more challenging database, the CNN-based system still mimics the trend we saw in human observers, i.e., that performance is better on the top 50% than on the bottom 50% of each of the three splits and is well above chance. For both the CNN and human categorization experiments, we run t-tests (see Table 6.1) which show that for both the group differences are statistically significant.

6.3.5 Experiments with Salience Weighted Contours

While we would expect that network performance would degrade when losing half the input pixels, the splits also reveal a significant bias in favor of our salience measures to support scene categorization. Can we exploit this bias to improve network performance when given the intact contours? To address this question, we carry out a second experiment where we explicitly encode salience measures for CNN by feeding different features into the R, G, and B color channels of the pre-trained network. We do this by using, in addition to the contour image channel, additional channels with the same contours weighted by our proposed salience measures, each of which is in the interval $[0, 1]$, as illustrated in Figure 6.8. These contour salience images replace the standard three-channel (R, G, B) inputs to the network. For all experiments, training is done on the feature maps generated by the new feature-coded images.

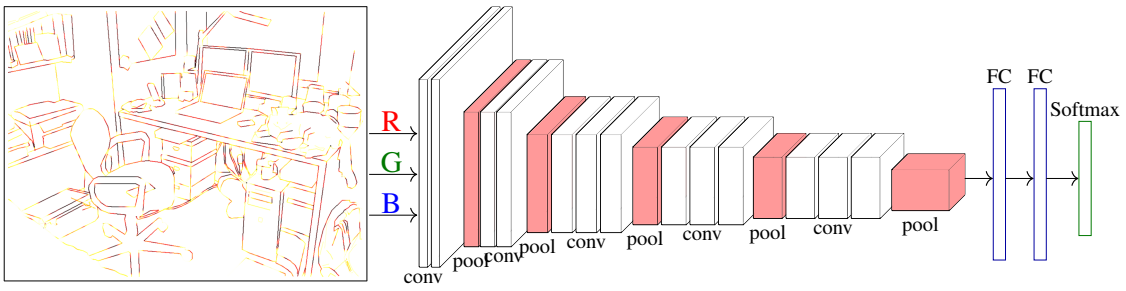


Figure 6.8: (Best viewed by zooming in on the PDF.) A schematic view of the VGG16 architecture with salience weighted contours used as the 3 input channels (see Table 6.2 for the specific sets of input channels.

As mentioned earlier, we initially used Dollar's edge detector output [39] and those results are reported for the Artist Scenes dataset and for MIT67 in Table 6.2 in [119].

Channels	Artist	MIT67
	VGG16	VGG16
Photos	98.95	64.87
Contours	90.53	42.80
Contours, Ribbon	93.49	45.24
Contours, Taper	94.71	43.66
Contours, Separation	93.91	43.89
Contours, Ribbon, Taper	95.02	45.36
Contours, Ribbon, Separation	95.89	**48.61**
Contours, Taper, Separation	**96.23**	47.18
Ribbon, Taper, Separation	94.38	44.82

Table 6.2: Top 1 level , on Artist Scenes and MIT67, with fine-tuning. TOP ROW: Results of the traditional R,G,B input configuration. OTHER ROWS: Combinations of intact scene contours, and scene contours weighted by our salience measures. Here, the MIT67 machine generated line drawings are based on Dollar's edge detection algorithm [39].

We then repeated those same set of experiments with our updated line drawings using Logical/Linear operators [68] and this time, we used VGG16-H (pre-trained on both Imagenet and Places365 [165]) in addition to VGG16 (pre-trained on Imagenet). In all the experiments the last two fully-connected layers of the pre-trained networks were fine-tuned using our feature-coded inputs, i.e., training was done on the feature maps provided by them. The results obtained using Logical/Linear operators of [68] for generating line drawings for the Artist Scenes dataset and for MIT67 are shown in Table 6.3.

First, it is noticeable that the Logical/Linear edge detection framework gives better results than Dollar's edge detection algorithm, presumably because of the importance of singularities and junctions for scene categorization. Second, it is apparent that with these salience weighted contour channels added, there is a consistent boost to the results obtained by using contours alone, independent of which machine-generated line drawing

Channels	Artist		MIT67	
	VGG16	VGG16-H	VGG16	VGG16-H
Photos	98.20	99.62	64.87	79.49
Contours	91.23	92.50	46.92	60.73
Contours, Ribbon	93.46	94.16	48.55	61.10
Contours, Taper	93.10	95.06	49.84	63.32
Contours, Separation	94.63	96.56	49.61	62.54
Contours, Ribbon, Taper	94.85	96.61	51.32	62.96
Contours, Ribbon, Separation	95.42	**98.40**	53.21	64.25
Contours, Taper, Separation	**96.82**	97.93	**54.17**	**65.79**
Ribbon, Taper, Separation	95.74	95.96	52.52	63.48

Table 6.3: Top 1 level performance in a 3-channel configuration, on the Artist Scenes and MIT67 databases, with fine-tuning. TOP ROW: Results of the traditional R,G,B input configuration where the original photos are used. OTHER ROWS: Combinations of intact scene contours, and scene contours weighted by our salience measures, where each letter stands for a specific input channel. Here, the MIT67 machine generated line drawings are based on the Logical/Linear edge detection framework [68].

algorithm is used. In all cases, the best performance boost comes from a combination of contours, ribbon or taper symmetry salience, and separation salience. We believe this is because taper between local contours as a perceptual salience measure is conceptually very close to our ribbon salience measure. Local separation salience, on the other hand, provides a more distinct and complementary perceptual cue for grouping.

For MIT67 the performance of 79.49% on photographs is consistent with that reported in [165]. Remarkably, 75% of this level of performance (a level of 60.73%) is obtained using *only* Logical/Linear line drawings. The overall performance goes up to 65.79% (or 82.8% of the performance on photographs) when using contours weighted by ribbon and separation salience. For MIT67, we have also compared the performance (fine-tuned) Hy-

brid1365_VGG on photographs (**79.49% top-1**) with photographs with contours, ribbon, and separation salience weighted contours overlayed (**82.05% top-1**). Thus, perceptually weighted contour features can boost overall performance as well.

Encouraged by the above results, we repeated the same experiment for the much more challenging Places365 dataset, but this time using just a pre-trained network and a linear SVM. For this dataset chance recognition performance would be at 1/365 or 0.27%. Our results using Dollar's method for machine generated line drawings, are shown in Table 6.4. Once again we see a clear and consistent trend of a benefit using salience

Channels	Places365 (Res50)
Photos	33.04
Contours	8.02
Contours, Ribbon	9.18
Contours, Taper	11.73
Contours, Separ	10.53
Contours, Ribbon, Taper	12.05
Contours, Ribbon, Separ	**14.23**
Contours, Taper, Separ	11.77
Ribbon, Taper, Separ	12.64

Table 6.4: Top 1 performance in a 3-channel configuration on Places365, with an off-the-shelf pre-trained network and a linear SVM (see text). The top row shows the results of the traditional R,G,B input configuration, while the others show combinations of intact scene contours and scene contours weighted by our salience measures. Here, the machine generated line drawings are based on Dollar's edge detection algorithm [39].

weighted contours as additional feature channels to the contours themselves, with the best performance gain coming from the addition of ribbon symmetry salience and separation salience. Finally, note that in the artist's line drawings, MIT67 and Places365 databases, the percentage of contour ink pixels over all the RGB pixels in the photographs, is only

7.44%, 8.75% and 8.32%, on average.

6.4 Discussion

Our experiments show that scene contours weighted by perceptually motivated contour salience measures can boost CNN-based scene categorization accuracy, despite the absence of color, texture and shading cues. Our work indicates that measures of contour grouping, which are simply functions of the contours themselves, are beneficial for scene categorization by computers, leading to recognition performance that is over 80% of the best reported results on the underlying photographs. Whereas this shape information is reflected in the images themselves, it does not appear to be directly learned by present state-of-the-art CNN-based scene recognition systems. Adding shape information computed via the medial axis outside of the CNNs improves scene categorization over the current state of the art. The obtained by our approach are in line with recent work by Geirhos et al. [57, 56], suggesting that using shape by CNNs for recognition/categorization is a promising direction for future work.

7

Conclusion

Artificial intelligence shows great promise in creating machines that can mimic human performance in complex tasks involving the processing of visual information. This includes rapid access to visual information from impoverished input such as silhouettes of 3D objects or line drawings of real-world scenes. Unfortunately, much less work has been done in the area of machine perception from such data. The current competitive methods use deep learning strategies which are applied directly to pixel intensities, thus combining both appearance information (color, texture) with geometric information encoded in the layout of contours. As such, they also require large amounts of data to train from - typically databases which include thousands or millions of photographs.

This doctoral thesis has focused on contour-based visual features for object recognition, environment mapping, and scene categorization. My work has shown that the encoding of contour geometry by adaptations of average outward flux-based skeletons, and the

deployment of suitable algorithms, provides a promising way to address all three problems. With the recent advances in computer vision and machine learning, many classical computer vision approaches are being abandoned by the robotics and vision community. This thesis revisits a very powerful representational tool and provides strong evidence of its applicability to unsolved problems, including view-based recognition. Further, it has demonstrated the use of classical Gestalt grouping cues, including measures for local symmetry or separation between contours, for boosting the performance of the present state of the art deep learning strategies for recognition and categorization.

7.1 Contributions

In the course of my doctoral work, I contributed novel research ideas to three classes of problems. A common theme in my approach to each class was the use of medial measures and the development, optimization, and deployment of appropriate algorithms rooted in AOF skeletons. The three classes of problems are summarized below.

View Based Object Recognition I addressed the problem of view-based 3D object recognition, which requires a selection of model object views against which to match a query view. Ideally, for this to be computationally efficient, such a selection should be sparse. To this end, I introduced a measure for skeletal branch simplification based on the uniqueness of a maximal inscribed disk to a skeletal branch. This relative area contribution measure is novel to the literature and is particularly simple to compute while being effective. The monotonicity property of this measure, i.e., the fact that it increases monotonically when moving away from a branch point, ensures that for each original skeletal

branch, at most one skeletal segment is retained. It allows one to associate each retained skeletal segment with the node of a graph, which we have dubbed the "Flux Graph". I have shown via experiments that the resulting representation is much less complex than earlier approaches, such as shock graphs. I then developed a strategy to partition the view sphere into regions within which the silhouette of a model object is qualitatively unchanged. This was accomplished by using AOF skeletons and skeletal matching to compute the pairwise similarity between two views. Broadly speaking, my research has shown the promise of view sphere partitioning for 3D recognition from sparse views, by the hierarchical application of a clustering algorithm on pairwise similarities computed between flux graphs.

2D Environment Mapping I have developed a novel online topological environment mapping algorithm that is based on the AOF skeleton. The algorithm works autonomously, i. e., the robot can automatically map the environment topologically at each time step and decide where to go next for further exploration. This is achieved by mapping the scanned environment as a map of nodes and edge paths (medial paths) using the AOF measure. The medial representation used in this approach is resistant to the measurement and sensor perturbations that occur in the gridmaps that the GMapping module produces. I have tested the algorithm both with real experiments using a Turtle Bot and with a stage simulator. I then sought to improve the stability of the proposed environment mapping by proposing a topological matching approach which can detect already visited environments, or map the regions that are not correctly reconstructed by the GMapping module. Our topological matching algorithm leverages the use of a spectral correspondence algorithm [90] that has been previously used in mesh correspondence problems. The novelty of our approach is

in the use of medial axis based radius information along with the medial segments themselves, in the distance measure that is used to compute the correspondences.

Contour Based Scene Categorization In my doctoral work, I have also developed AOF based algorithms to measure and characterize the role of different local geometric properties of contours, including symmetry, parallelism, and proximity for scene categorization and scene perception. The measures I have introduced have been used to separate the contour pixels in a given scene into the more salient and the less salient halves. We have demonstrated that human observers are better at categorizing scenes containing only the more salient halves in a rapid-categorization psychophysical experiment. We have also demonstrated that the use of Gestalt cues to weight contours according to their salience, improve benefits scene categorization by CNNs. Our experiments show that scene contours weighted by perceptually motivated contour salience measures can boost CNN-based scene categorization accuracy, despite the absence of color, texture and shading cues. I also show that the inclusion of medial-axis based contour salience weights leads to a further boost in the recognition of real scene photographs. Whereas the shape information is reflected in the images themselves, it does not appear to be directly learned by present state-of-the-art CNN-based scene recognition systems. Adding shape information computed on the medial axis outside of the CNNs improves scene categorization above the current state of the art. To evaluate the methods I developed for contour-based scene abstraction and categorization, I also significantly extended the contour databases previously used for bench-marking computer vision systems used for such tasks. Notably, existing databases of line drawings contain hundreds or thousands of images. I generated

a database containing millions of line drawings, created from photographs of complex scenes in the Places dataset [114, 165] by using the Logical/Linear edge detection framework of Iverson and Zucker [68]. My research has demonstrated the promise of perceptual organization principles from human vision in improving the capabilities of computer vision-based scene categorization and recognition systems (see Section 6.3.5 from Chapter 6).

7.2 Future Research Directions

There are many potential avenues for future work related to the ideas and results in this thesis. A discussion of what I consider to be some of the most fruitful avenues for exploration appears below.

3D object detection is a fundamental challenge for many modern artificial intelligence systems such as automated driving, autonomous robot perception sensing, and intelligent detection and surveillance systems. In work presented in this thesis, we provide an approach to pick candidate views intelligently for view-based recognition problems. Our work suggests several fruitful directions for further research, having to do with the use of precomputed view sphere partitions in online recognition scenarios. With the advances that have been made in the production of vision sensors in recent years, recognition of real 3D objects seems more possible than ever. Using the algorithm presented in Chapter 3, one could apply these ideas in a similar scenario to recognize 3D real-world objects from their outlines. With the rich information contained in the silhouette of a 3D subject, enterprise systems that process large scale of information can take advantage of view-based

recognition strategies to handle sparse views.

The approach to online topological mapping presented in Chapter 4 introduces a unique opportunity for the robotics community to utilize our proof of concept ideas and explore several different possibilities. In our work, we have shown that sensor-based environment mapping algorithms do not need to analyze large, complex data to guide a robot in an unseen environment, albeit, the sensors have seen a lot of advances in recent years and the amount of information that can be captured through these sensors is growing more than ever. Our autonomous algorithm is a robust mapping strategy that could be easily utilized. Our algorithm can be applied through a variety of sensors without the need to handle of complex visual features across the mapped environment. The framework presented here can be tested for various types of environments/sensors. Investigating scenarios where the autonomous environment mapping should be executable with minimal computational resources, could be another direction for our work. In addition, this work can be extended to map 3D environments such as those which occur underwater or in the air. By expanding the experiments, we could compare the mapped environments with other existing methods for different scenarios. Finally, the ideas presented in this work can be utilized for problems with different mapping modules (other than GMapping).

In recent work by Geirhos et al. [57], the authors show that CNNs that are used to recognize object classes are biased towards learning texture in visual inputs that learn complex representations of object shapes. Our recent work in scene categorization from line drawings is beginning to demonstrate that there is much information in shape and contour geometry that is not exploited by patch-based CNN systems, complementing but

also resonating with the ideas of [57]. We have presented results to showcase the importance of contours and perceptually weighted contours. One future direction for this work is that of identifying and formulating a complete list of principles that guide the perceptual organization of local contour elements, providing support to the human visual system in the understanding of cluttered, real-world scenes in a more comprehensive way. Preliminary work along these lines has been reported in Chapter 5 of this thesis. Extending these ideas, assessing the viability of these principles for facilitating the categorization of complex real-world environments in computer vision systems, is a key next step to consider. Interpreting the role of perceptual grouping cues in tasks such as recognition or categorization of objects or entire scenes in biological and artificial vision systems could be a promising direction for future work.

Bibliography

[1] Gerald J Agin and Thomas O Binford. Computer description of curved objects. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pages 629–640. Morgan Kaufmann Publishers Inc., 1973.

[2] Dominique Attali, Jean-Daniel Boissonnat, and Herbert Edelsbrunner. Stability and computation of medial axes - a State-of-the-Art report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization, pages 109–125. Springer Berlin Heidelberg, 2009.

[3] Fred Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183, 1954.

[4] Shuang Bai. Growing random forest on deep convolutional neural networks for scene categorization. *Expert Systems with Applications*, 71:279–287, 2017.

[5] X. Bai and L. J. Latecki. Path similarity skeleton graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1282–1292, July 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.70769.

[6] Xiang Bai, Xingwei Yang, Deguang Yu, and Longin Jan Latecki. Skeleton-based shape classification using path similarity. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(04):733–746, 2008. doi: 10.1142/S0218001408006405. URL https://doi.org/10.1142/S0218001408006405.

[7] Richard Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16 (1):87–90, 1958.

[8] Serge Belongie and Jitendra Malik. Matching with shape contexts. In *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, pages 20–26, 2000.

[9] Daniel Berman, Julie D. Golomb, and Dirk B. Walther. Scene content is predominantly conveyed by high spatial frequencies in scene-selective visual cortex. *PLOS ONE*, 12(12):1–16, 12 2017. doi: 10.1371/journal.pone.0189828. URL https://doi.org/10.1371/journal.pone.0189828.

[10] Irving Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image processing*, 32(1):29–73, 1985.

[11] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological Review*, 94(2):115, 1987.

[12] Irving Biederman, Robert J Mezzanotte, and Jan C Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology*, 14(2):143–177, 1982.

[13] Thomas Binford. Visual perception by computer. In *IEEE Conference of Systems and Control*, pages 1489–1496, 1971.

[14] Colin Blakemore and Brian Hague. Evidence for disparity detecting neurones in the human visual system. *The Journal of Physiology*, 225(2):437–455, 1972.

[15] Harry Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual*, 19(5):362–380, 1967.

[16] Harry Blum. Discussion paper: A geometry for biology. *Annals of the New York Academy of Sciences*, 231(1):19–30, 1974.

[17] Harry Blum and Roger N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167 – 180, 1978. ISSN 0031-3203. doi: 10.1016/0031-3203(78)90025-0. URL http://www.sciencedirect.

com/science/article/pii/0031320378900250. The Proceedings of the IEEE Computer Society Conference.

[18] H.J. Blum. Biological shape and visual science (part 1). *Journal of Theoretical Biology*, 38:205–287, 1973.

[19] Michael Brady and Haruo Asada. Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61, 1984.

[20] Emma Brunskill, Thomas Kollar, and Nicholas Roy. Topological mapping using spectral clustering and classification. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3491–3496. IEEE, 2007.

[21] Christina A. Burbeck and Stephen M. Pizer. Object representation by cores: Identifying and representing primitive spatial regions. *Vision Research*, 35(13):1917 – 1930, 1995. ISSN 0042-6989. doi: https://doi.org/10.1016/0042-6989(94)00286-U. URL http://www.sciencedirect.com/science/article/pii/004269899400286U.

[22] Charles F Cadieu, Ha Hong, Daniel LK Yamins, Nicolas Pinto, Diego Ardila, Ethan A Solomon, Najib J Majaj, and James J DiCarlo. Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Computational Biology*, 10(12):e1003963, 2014. URL https://doi.org/10.1371/journal.pcbi.1003963.

[23] J. Canny, R. Seidel, and Z. Gigus. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(06):542–551, jun 1991. ISSN 0162-8828. doi: 10.1109/34.87341.

[24] Moses W Chan, Adam K Stevenson, Yunfeng Li, and Zygmunt Pizlo. Binocular shape constancy from novel views: The role of a priori constraints. *Perception & Psychophysics*, 68(7):1124–1139, 2006.

[25] Raja Chatila and Jean-Paul Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 138–145. IEEE, 1985.

[26] Frédéric Chazal and André Lieutier. The "λ-medial axis". *Graph. Models*, 67(4): 304–331, July 2005.

[27] Frédéric Chazal and Rémi Soufflet. Stability and finiteness properties of medial axis and skeleton. *Journal of Dynamical and Control Systems*, 10(2):149–170, 2004.

[28] Heeyoung Choo and Dirk B Walther. Contour junctions underlie neural representations of scene categories in high-level human visual cortex. *NeuroImage*, 135: 32–44, 2016.

[29] H. Choset and J. Burdick. Sensor based planning. ii. incremental construction of the generalized voronoi graph. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, volume 2, pages 1643–1648 vol.2, May 1995. doi: 10.1109/ROBOT.1995.525510.

[30] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, April 2001. ISSN 1042-296X. doi: 10.1109/70.928558.

[31] Howie Choset. Incremental construction of the generalized voronoi diagram, the generalized voronoi graph, and the hierarchical generalized voronoi graph. In *Proc. First CGC Workshop on Computational Geometry*, 1997.

[32] Howie M Choset, Keiji Nagatani, and Alfred A Rizzi. Sensor-based planning: using a honing strategy and local map method to implement the generalized voronoi graph. In *Mobile Robots XII*, volume 3210, pages 72–84. International Society for Optics and Photonics, 1998.

[33] Luciano da Fontoura Da Costa and Roberto Marcondes Cesar Jr. *Shape analysis and classification: theory and practice*. CRC Press, Inc., 2000.

[34] Christopher M Cyr and Benjamin B Kimia. A similarity-based aspect-graph approach to 3d object recognition. *International Journal of Computer Vision*, 57(1): 5–22, 2004.

[35] Claudia Damiano, John Wilder, and Dirk B Walther. Mid-level feature contributions to category-specific gaze guidance. *Attention, Perception, & Psychophysics*, pages 1–12, 2018.

[36] Arnaud Delorme, Guillaume Richard, and Michele Fabre-Thorpe. Ultra-rapid categorisation of natural scenes does not rely on colour cues: a study in monkeys and humans. *Vision Research*, 40(16):2187–2200, 2000.

[37] Sven J. Dickinson. *The Evolution of Object Categorization and the Challenge of Image Abstraction*, pages 1–37. Cambridge University Press, 2009. doi: 10.1017/CBO9780511635465.002.

[38] Pavel Dimitrov, James N Damon, and Kaleem Siddiqi. Flux invariants for shape. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–835. IEEE, 2003.

[39] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1848, 2013.

[40] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Using multiple markers in graph exploration. In *Proc. Symposium on Advances in Intelligent Robotics Systems: Conf. on Mobile Robotics*, 1989.

[41] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865, 1991.

[42] Gregory Dudek, Paul Freedman, and Souad Hadjres. Using local information in a non-local way for mapping graph-like worlds. In *International Joint Conference on Artificial Intelligence*, pages 1639–1647, 1993.

[43] Shimon Edelman. *Representation and recognition in vision*. MIT press, 1999.

[44] D. W. Eggert, K. W. Bowyer, C. R. Dyer, H. I. Christensen, and D. B. Goldgof. The scale space aspect graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1114–1130, Nov 1993. ISSN 0162-8828. doi: 10.1109/34.244674.

[45] James H Elder and Steven W Zucker. Computing contour closure. In *European Conference on Computer Vision*, pages 399–412. Springer, 1996.

[46] Russell Epstein and Nancy Kanwisher. A cortical representation of the local visual environment. *Nature*, 392(6676):598, 1998.

[47] Ricardo Fabbri, Leandro Farias Estrozi, and L Da F Costa. On voronoi diagrams and medial axes. *Journal of Mathematical Imaging and Vision*, 17(1):27–40, 2002.

[48] Olivier Faugeras, Joe Mundy, Narendra Ahuja, Charles Dyer, Alex Pentland, Ramesh Jain, Katsushi Ikeuchi, and Kevin Bowyer. Why aspect graphs are not (yet) practical for computer vision. *CVGIP: Image Understanding*, 55(2):212–218, 1992.

[49] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[50] Jacob Feldman and Manish Singh. Information along contours and object boundaries. *Psychological review*, 112(1):243, 2005.

[51] Jacob Feldman and Manish Singh. Bayesian estimation of the shape skeleton. *Proceedings of the National Academy of Sciences*, 103(47):18014–18019, 2006.

[52] Robert Fergus, Pietro Perona, and Andrew Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3):273–303, 2007.

[53] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, 67(2):159–188, 2006.

[54] Chaz Firestone and Brian J Scholl. "Please tap the shape, anywhere you like": Shape skeletons in human vision revealed by an exceedingly simple measure. *Psychological Science*, page 0956797613507584, 2014.

[55] Santiago Garrido, Luis Moreno, Dolores Blanco, and Piotr Jurewicz. Path planning for mobile robot navigation using voronoi diagram and fast marching. *Int. J. Robot. Autom*, 2(1):42–64, 2011.

[56] Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7538–7550. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7982-generalisation-in-humans-and-deep-neural-networks.pdf.

[57] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

[58] Ziv Gigus and Jitendra Malik. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):113–122, 1990.

[59] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[60] Leo J Grady and Jonathan Polimeni. *Discrete calculus: Applied analysis on graphs for computational science*. Springer Science & Business Media, 2010.

[61] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *IEEE International Conference on Robotics and Automation*, pages 2432–2437. IEEE, 2005.

[62] Umut Güçlü and Marcel A. J. van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, 2015. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.5023-14.2015. URL http://www.jneurosci.org/content/35/27/10005.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[64] D.D. Hoffman and W.A. Richards. Parts of recognition. *Cognition*, 18(1):65–96, 1984.

[65] Donald D Hoffman and Manish Singh. Salience of visual parts. *Cognition*, 63(1):29 – 78, 1997. ISSN 0010-0277. doi: https://doi.org/10.1016/S0010-0277(96)00791-3. URL http://www.sciencedirect.com/science/article/pii/S0010027796007913.

[66] Shin Hoo-Chang, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset

characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35 (5):1285, 2016.

[67] David H Hubel and TN Wiesel. Shape and arrangement of columns in cat's striate cortex. *The Journal of Physiology*, 165(3):559–568, 1963.

[68] Lee A. Iverson and Steven W. Zucker. Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):982–996, 1995.

[69] Gaetano Kanizsa, Walter Gerbino, et al. Convexity and symmetry in figure-ground organization. *Vision and Artifact*, pages 25–32, 1976.

[70] Nancy Kanwisher, Josh McDermott, and Marvin M Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, 17(11):4302–4311, 1997.

[71] Robert A. Katz and Stephen M. Pizer. Untangling the blum medial axis transform. *International Journal of Computer Vision*, 55(2-3):139–153, November 2003. ISSN 0920-5691. doi: 10.1023/A:1026183017197. URL https://doi.org/10.1023/A:1026183017197.

[72] Philip J Kellman and Thomas F Shipley. A theory of visual interpolation in object perception. *Cognitive Psychology*, 23(2):141–221, 1991.

[73] John M Kennedy and Ramona Domander. Shape and contour: The points of maximum change are least useful for recognition. *Perception*, 14(3):367–370, 1985.

[74] Benjamin B. Kimia, Allen R. Tannenbaum, and Steven W. Zucker. Shapes, shocks, and deformations i: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15(3):189–224, Jul 1995. ISSN 1573-1405. doi: 10.1007/BF01451741. URL https://doi.org/10.1007/BF01451741.

[75] J. J. Koenderink and A. J. van Doorn. The singularities of the visual mapping. *Biological Cybernetics*, 24(1):51–59, Mar 1976. ISSN 1432-0770. doi: 10.1007/BF00365595. URL https://doi.org/10.1007/BF00365595.

[76] Jan J Koenderink and Andrea J van Doorn. The internal representation of solid shape with respect to vision. *Biological ybernetics*, 32(4):211–216, 1979.

[77] Kurt Koffka. Perception: An introduction to the Gestalt-theorie. *Psychological Bulletin*, 19(10):531–585, 1922.

[78] Kurt Koffka. *Principles of gestalt psychology,*. Harcourt, Brace and Company, 1935.

[79] W Köhler. Gestalt Psychology (1929). *New York, NY: Liveright*, 1947.

[80] Ilona Kovács and Bela Julesz. Perceptual sensitivity maps within globally defined visual shapes. *Nature*, 370(6491):644, 1994.

[81] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8(1-2):47–63, 1991.

[82] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4845–4851. IEEE, 2004.

[83] Michael Leyton. Symmetry-curvature duality. *Computer Vision, Graphics, and Image Processing*, 38(3):327 – 341, 1987. ISSN 0734-189X. doi: https://doi.org/10.1016/0734-189X(87)90117-4. URL http://www.sciencedirect.com/science/article/pii/0734189X87901174.

[84] Michael Leyton. A process-grammar for shape. *Artificial Intelligence*, 34(2):213–247, 1988.

[85] Yunfeng Li and Zygmunt Pizlo. Perception of 3d shapes from line drawings. *Journal of Vision*, 8(6):452–452, 2008.

[86] Haibin Ling and David W Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, 2007.

[87] Norah K Link and Steven W Zucker. Corner detection in curvilinear dot grouping. *Biological Cybernetics*, 59(4-5):247–256, 1988.

[88] Brad Lisien, Deryck Morales, David Silver, George Kantor, Ioannis M. Rekleitis, and Howie Choset. The hierarchical atlas. *IEEE Transactions on Robotics*, 21(3): 473–481, June 2005.

[89] Yanxi Liu, Hagit Hel-Or, Craig S Kaplan, Luc Van Gool, et al. Computational symmetry in computer vision and computer graphics. *Foundations and Trends® in Computer Graphics and Vision*, 5(1–2):1–195, 2010.

[90] Herve Lombaert, Leo Grady, Jonathan R Polimeni, and Farida Cheriet. Focusr: Feature oriented correspondence using spectral regularization–a method for precise surface matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2143–2160, 2013.

[91] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, Sep. 1999. doi: 10.1109/ICCV.1999.790410.

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

[93] Bart Machilsen, Maarten Pauwels, and Johan Wagemans. The role of vertical mirror symmetry in visual shape detection. *Journal of Vision*, 9(12):11–11, 2009.

[94] Diego Macrini, Kaleem Siddiqi, and Sven Dickinson. From skeletons to bone graphs: Medial abstraction for object recognition. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[95] Rafael Malach, JB Reppas, RR Benson, KK Kwong, H Jiang, WA Kennedy, PJ Ledden, TJ Brady, BR Rosen, and RB Tootell. Object-related activity revealed by functional magnetic resonance imaging in human occipital cortex. *Proceedings of the National Academy of Sciences*, 92(18):8135–8139, 1995.

[96] David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978.

[97] E. Masehian, M. R. Amin-Naseri, and S. E. Khadem. Online motion planning using incremental construction of medial axis. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 2928–2933 vol.3, Sep. 2003. doi: 10.1109/ROBOT.2003.1242040.

[98] H Murase and SK Nayar. Learning and recognition of 3D objects from appearance. In *Qualitative Vision, 1993., Proceedings of IEEE Workshop on*, pages 39–50, June 1993.

[99] Hiroshi Murase and Shree K Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[100] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[101] A Nathan. College geometry: An introduction to the modern geometry of the triangle and the circle, 1952.

[102] R.L. Ogniewicz. Skeleton-space: a multiscale shape description combining region and boundary information. In *Computer Vision and Pattern Recognition, 1994.*

Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, pages 746 –751, June 1994. doi: 10.1109/CVPR.1994.323891.

[103] Robert L Ogniewicz. *Discrete Voronoi Skeletons*. PhD thesis, Diss. Techn. Wiss. ETH Zürich, Nr. 9876, 1992. Ref.: O. Kübler; Korref.: T. Pun, 1992.

[104] Aude Oliva and Philippe G Schyns. Diagnostic colors mediate scene recognition. *Cognitive Psychology*, 41(2):176–210, 2000.

[105] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[106] Alex P Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986.

[107] David I Perrett and Mike W Oram. Neurophysiology of shape processing. *Image and Vision Computing*, 11(6):317–333, 1993.

[108] Sylvain PetitJean, Jean Ponce, and David J Kriegman. Computing exact aspect graphs of curved objects: Algebraic surfaces. *International Journal of Computer Vision*, 9(3):231–255, 1992.

[109] Zygmunt Pizlo. *Making a machine that sees like us*. Oxford University Press (UK), 2014.

[110] Harry Plantinga and Charles R Dyer. Visibility, occlusion, and the aspect graph. *International Journal of Computer Vision*, 5(2):137–160, 1990.

[111] William H Plantinga and Charles R Dyer. An algorithm for constructing the aspect graph. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 1985.

[112] Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman. *Toward category-level object recognition*, volume 4170. Springer, 2007.

[113] Mary C Potter and Ellen I Levy. Recognition memory for a rapid sequence of pictures. *Journal of Experimental Psychology*, 81(1):10, 1969.

[114] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009.

[115] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[116] M. Rezanejad, B. Samari, I. Rekleitis, K. Siddiqi, and G. Dudek. Robust environment mapping using flux skeletons. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5700–5705, Sep. 2015. doi: 10.1109/IROS.2015.7354186.

[117] Morteza Rezanejad and Kaleem Siddiqi. Flux graphs for 2d shape analysis. In Sven J. Dickinson and Zygmunt Pizlo, editors, *Shape Perception in Human and Computer Vision*, Advances in Computer Vision and Pattern Recognition, pages 41–54. Springer London, 2013. ISBN 978-1-4471-5194-4. doi: 10.1007/978-1-4471-5195-1_3. URL http://dx.doi.org/10.1007/978-1-4471-5195-1_3.

[118] Morteza Rezanejad and Kaleem Siddiqi. View sphere partitioning via flux graphs boosts recognition from sparse views. *Frontiers in ICT*, 2:24, 2015. ISSN 2297-198X. doi: 10.3389/fict.2015.00024. URL http://journal.frontiersin.org/article/10.3389/fict.2015.00024.

[119] Morteza Rezanejad, Gabriel Downs, John Wilder, Dirk B Walther, Allan Jepson, Sven Dickinson, and Kaleem Siddiqi. Scene categorization from contours: Medial axis based salience measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[120] Morteza Rezanejad, Gabriel Downs, John Wilder, Dirk B Walther, Allan Jepson, Sven Dickinson, and Kaleem Siddiqi. Gestalt-based contour weights improve scene categorization by cnns. In *Conference on Cognitive Computational Neuroscience*, 2019.

[121] Murray B Sachs, Jacob Nachmias, and John G Robson. Spatial-frequency channels in human vision. *JOSA*, 61(9):1176–1186, 1971.

[122] Edward B Saff and A BJ Kuijlaars. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19(1):5–11, 1997.

[123] Sudeep Sarkar and Kim L Boyer. Perceptual organization in computer vision: status, challenges, and potential. *Computer Vision and Image Understanding*, 76(1): 1–5, 1999.

[124] Silvio Savarese and Li Fei-Fei. 3d generic object categorization, localization and pose estimation. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, October 2007.

[125] Francesco Savelli. Loop-closing and planarity in topological map-building. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 1511–1517, 2004.

[126] Michel Schmitt. Some examples of algorithms analysis in computational geometry by means of mathematical morphological techniques. In *Geometry and Robotics*, Lecture Notes in Computer Science, pages 225–246. Springer Berlin Heidelberg, 1989.

[127] Thomas B Sebastian, Philip N Klein, and Benjamin B Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004.

[128] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[129] Damian J Sheehy, Cecil G Armstrong, and Desmond J Robinson. Shape description by medial surface construction. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):62–72, 1996.

[130] Evan C Sherbrooke, Nicholas M Patrikalakis, and Erik Brisson. An algorithm for the medial axis transform of 3d polyhedral solids. *IEEE transactions on visualization and computer graphics*, 2(1):44–61, 1996.

[131] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[132] Kaleem Siddiqi and Benjamin B Kimia. Parts of visual form: Computational aspects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3): 239–251, 1995.

[133] Kaleem Siddiqi and Stephen Pizer. *Medial representations: mathematics, algorithms and applications*, volume 37. Springer Science and Business Media, 2008.

[134] Kaleem Siddiqi, Kathryn J Tresness, and Benjamin B Kimia. Parts of visual form: Psychophysical aspects. *Perception*, 25(4):399–424, 1996. doi: 10.1068/p250399. URL https://doi.org/10.1068/p250399. PMID: 8817619.

[135] Kaleem Siddiqi, Sylvain Bouix, Allen Tannenbaum, and Steven W Zucker. The hamilton-jacobi skeleton. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 828–834. IEEE, 1999.

[136] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson, and Steven W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35 (1):13–32, 1999. ISSN 0920-5691. URL http://dx.doi.org/10.1023/A:1008102926703.

[137] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. URL http://arxiv.org/abs/1409.1556.

[138] Manish Singh and Donald D. Hoffman. 13 - part-based representations of visual shape and implications for visual cognition. In Thomas F. Shipley and Philip J. Kellman, editors, *From Fragments to Objects*, volume 130 of *Advances in Psychology*, pages 401 – 459. North-Holland, 2001. doi: https://doi.org/10.1016/S0166-4115(01)80033-9. URL http://www.sciencedirect.com/science/article/pii/S0166411501800339.

[139] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015.

[140] Joachim S Stahl and Song Wang. Globally optimal grouping for symmetric closed boundaries by combining boundary and region information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):395–411, 2008.

[141] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[142] Simon Thorpe, Denis Fize, Catherine Marlot, et al. Speed of processing in the human visual system. *Nature*, 381(6582):520–522, 1996.

[143] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[144] Antonio Torralba and Aude Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3):391–412, 2003.

[145] Stephen Tully, George Kantor, Howie Choset, and Felix Werner. A multi-hypothesis topological SLAM approach for loop closing on edge-ordered graphs. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4943–4948, 2009.

[146] Rufin VanRullen and Simon J Thorpe. Is it a bird? is it a plane? Ultra-rapid visual categorisation of natural and artifactual objects. *Perception*, 30(6):655–668, 2001.

[147] Johan Wagemans. Skewed symmetry: a nonaccidental property used to perceive visual forms. *Journal of Experimental Psychology: Human Perception and Performance*, 19(2):364, 1993.

[148] Johan Wagemans. Characteristics and models of human symmetry detection. *Trends in Cognitive Sciences*, 1(9):346–352, 1997.

[149] Johan Wagemans, James H Elder, Michael Kubovy, Stephen E Palmer, Mary A Peterson, Manish Singh, and Rüdiger von der Heydt. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization. *Psychological Bulletin*, 138(6):1172, 2012.

[150] Dirk B Walther and Dandan Shen. Nonaccidental properties underlie human categorization of complex natural scenes. *Psychological science*, page 0956797613512662, 2014.

[151] Dirk B Walther, Barry Chai, Eamon Caddigan, Diane M Beck, and Li Fei-Fei. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108(23):9661–9666, 2011.

[152] Hui Wang, Michael Jenkin, and Patrick Dymond. Enhancing exploration in graph-like worlds. In *2008 Canadian Conference on Computer and Robot Vision*, pages 53–60. IEEE, 2008.

[153] Felix Werner, Frederic Maire, Joaquin Sitte, Howie Choset, Stephen Tully, and George Kantor. Topological slam using neighbourhood information of places. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4937–4942. IEEE, 2009.

[154] Max Wertheimer. *Laws of organization in perceptual forms.* Kegan Paul, Trench, Trubner & Company, 1938.

[155] Felix A Wichmann, Jan Drewes, Pedro Rosas, and Karl R Gegenfurtner. Animal detection in natural scenes: critical features revisited. *Journal of Vision*, 10(4):6–6, 2010.

[156] John Wilder, Jacob Feldman, and Manish Singh. Superordinate shape classification using natural shape statistics. *Cognition*, 119(3):325–340, 2011.

[157] John Wilder, Jacob Feldman, and Manish Singh. The role of shape complexity in the detection of closed contours. *Vision research*, 126:220–231, 2016.

[158] John Wilder, Sven Dickinson, Allan Jepson, and Dirk B Walther. Spatial relationships between contours impact rapid scene classification. *Journal of Vision*, 18(8): 1–1, 2018.

[159] John Wilder, Morteza Rezanejad, Sven Dickinson, Kaleem Siddiqi, Allan Jepson, and Dirk B. Walther. Local contour symmetry facilitates scene categorization. *Cognition*, 182:307 – 317, 2019. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2018.09.014. URL http://www.sciencedirect.com/science/article/pii/S0010027718302506.

[160] Ying Nian Wu, Zhangzhang Si, Haifeng Gong, and Song-Chun Zhu. Learning active basis model for object detection and recognition. *International Journal of Computer Vision*, 90(2):198–235, 2010.

[161] Yangsheng Xu, Raju Mattikalli, and Pradeep Khosla. Motion planning using medial axis. *IFAC Proceedings Volumes*, 25(28):135 – 140, 1992. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)49480-8. URL http://www.sciencedirect.com/science/article/pii/S1474667017494808. IFAC Workshop on Intelligent Manufacturing Systems (IMS'92), Dearborn, MI, USA, 1-2 October 1992.

[162] Yao Xu, Bo Wang, Wenyu Liu, and Xiang Bai. Skeleton graph matching based on critical points using path similarity. In *Computer Vision - ACCV 2009*, Lecture Notes in Computer Science, pages 456–465. Springer Berlin Heidelberg, 1 January 2010.

[163] Xingwei Yang, Xiang Bai, Deguang Yu, and Longin Jan Latecki. Shape classification based on skeleton path similarity. In *Energy Minimization Methods in Com-*

puter Vision and Pattern Recognition, Lecture Notes in Computer Science, pages 375–386. Springer Berlin Heidelberg, 1 January 2007.

[164] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2005.

[165] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.