

SOURCE-AGGREGATED-POISSON WITH APPLICATIONS TO GROUPWISE  
SHAPE ANALYSIS AND MESH SEGMENTATION

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT GENÇTAV

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
COMPUTER ENGINEERING

SEPTEMBER 2018





Approval of the thesis:

**SOURCE-AGGREGATED-POISSON WITH APPLICATIONS TO  
GROUPWISE SHAPE ANALYSIS AND MESH SEGMENTATION**

submitted by **MURAT GENÇTAV** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar  
Dean, Graduate School of **Natural and Applied Sciences**

\_\_\_\_\_

Prof. Dr. Halit Oğuztüzün  
Head of Department, **Computer Engineering**

\_\_\_\_\_

Prof. Dr. Sibel Tari  
Supervisor, **Computer Engineering Department, METU**

\_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. M. Kemal LEBLEBİCİOĞLU  
Electrical and Electronics Engineering Department, METU

\_\_\_\_\_

Prof. Dr. Sibel Tari  
Computer Engineering Department, METU

\_\_\_\_\_

Prof. Dr. Volkan Atalay  
Computer Engineering Department, METU

\_\_\_\_\_

Prof. Dr. Uğur Gündükbay  
Computer Engineering Department, Bilkent University

\_\_\_\_\_

Assist. Prof. Dr. Rüyam Acar  
Computer Engineering Department, Okan University

\_\_\_\_\_

**Date:**

\_\_\_\_\_

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: MURAT GENÇTAV

Signature :

## ABSTRACT

### **SOURCE-AGGREGATED-POISSON WITH APPLICATIONS TO GROUPWISE SHAPE ANALYSIS AND MESH SEGMENTATION**

Gençtav, Murat

Ph.D., Department of Computer Engineering

Supervisor : Prof. Dr. Sibel Tari

September 2018, 148 pages

By computing multiple solutions to Poisson’s equation with varying source functions within the shape and aggregating those solutions, we obtain a novel function for shape analysis, which we call Source-Aggregated-Poisson, or SAP. Despite the local computations, by means of specially designed source functions, our model mimics the part-coding behavior of a previous nonlocal model. We show that SAP is robust under geometric transformations and nuisance factors including topological distortions, pose changes, and occlusions. Using SAP, we address shape analysis problems in two and three dimensions. Toward this end, firstly, we exploit the evolution of its level curves and extract a probabilistic representation of shape decomposition hierarchy. Then, in the context of a groupwise shape analysis task, we demonstrate how such a probabilistic structure enables us to select the task-dependent optimum from the set of possible hierarchies. Finally, we devise an unsupervised mesh segmentation algorithm which utilizes SAP after projecting it to the surface mesh. Benchmark evaluation shows that the algorithm performs the best among the unsupervised algorithms and even performs comparable to supervised and groupwise segmentation methods.

Keywords: Poisson's Equation, Shape Analysis, Hierarchical Shape Decomposition,  
Groupwise Analysis, 3D Mesh Segmentation

## ÖZ

### DEĞİŞEN KAYNAKLI BİRLEŞİK POISSON VE ŞEKİL GRUPLARININ ANALİZİ İLE YÜZEY AĞLARININ BÖLÜTLENMESİNE UYGULANMASI

Gençtav, Murat

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Sibel Tarı

Eylül 2018 , 148 sayfa

Bu çalışmada Poisson denkleminin değişen kaynak fonksiyonlarıyla elde edilen çözümleri bir araya getirilerek şekil analizi için değişen kaynaklı birleşik Poisson adını verdiğimiz bir fonksiyon önerilmiştir. Temel hesaplama modelimiz yerel olmakla birlikte özel olarak dizayn edilmiş kaynak fonksiyonları sayesinde daha eski yerel olmayan bir modelin parça kodlama davranışını taklit edebilmektedir. Önerilen fonksiyon geometrik dönüşümler ile topolojik bozulmalar, poz değişimi ve kapatmalar gibi faktörlere karşı güvenilirdir. Fonksiyon iki ve üç boyutta şekil analizi problemlerinde kullanılmıştır. Bu doğrultuda, önce, fonksiyonun seviye eğrilerinin evriminden faydalanılarak şekil bölünme hiyerarşisinin olasılıksal bir temsili çıkarılmıştır. Sonrasında, şekillerin grup halinde analizi uygulamasında, çıkarılan bu olasılıksal gösterimin olasılıksal hiyerarşiler kümesinden uygulamaya en uygun olanı seçmeyi mümkün kılması gösterilmiştir. Son olarak, fonksiyonu tanımlı olduğu şekil alanından şekil çevresine yansıtarak kullanan güdümsüz yüzey ağı bölütleme algoritması geliştirilmiş ve kıyaslamalı değerlendirmede aynı türdeki algoritmalar arasında en iyi ve hatta güdümlü ve grup bölütleme algoritmalarına yakın başarımlar elde edilmiştir.

Anahtar Kelimeler: Poisson Denklemi, Şekil Analizi, Hiyerarşik Şekil Parçalama, Ortak Analiz, 3B Yüzey Ağı Bölütleme

*To my dear wife*

## **ACKNOWLEDGMENTS**

I am profoundly grateful to my advisor Prof. Dr. Sibel Tarı for her invaluable guidance, support and encouragement throughout this study. It is a great honor to be your student and to have met you. And thank you for your faith in me, for the opportunities you provided, and for sharing your wisdom that helped me improve. I learned so much and I owe it all to you.

To my dear wife Aslı: It would be impossible to get this work done without your love and emotional support that helped me through hard times. So glad I have you.

I am also grateful to my family. I always felt their unfailing support with me.

Special thanks goes to the members of the thesis committee, for their contributions that improved the quality of the thesis.

I want to express my gratitude to Middle East Technical University, especially to Computer Engineering Department for providing me the opportunities to pursue my Ph.D. and to work as a teaching assistant.

I also gratefully acknowledge the support of The Scientific and Technological Research Council of Turkey (TUBITAK) under scholarship program 2211/A and under Grant Numbers 112E208 and 114E204.



## TABLE OF CONTENTS

ABSTRACT . . . . .	v
ÖZ . . . . .	vii
ACKNOWLEDGMENTS . . . . .	x
TABLE OF CONTENTS . . . . .	xi
LIST OF TABLES . . . . .	xiv
LIST OF FIGURES . . . . .	xv
LIST OF ALGORITHMS . . . . .	xx
LIST OF ABBREVIATIONS . . . . .	xxi
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Shape and Perception . . . . .	1
1.2 Goal and Scope of the Study . . . . .	2
1.3 Motivating Applications of Shape Analysis . . . . .	2
1.4 Contributions and Thesis Organization . . . . .	3
2 A LOCAL ALTERNATIVE TO A PREVIOUS NONLOCAL MODEL FOR SHAPE ANALYSIS . . . . .	5
2.1 Review of the Previous Model [66] . . . . .	5

2.2	Mimicking the Previous Nonlocal Model Via Local Computations . . . . .	7
2.3	Computational Aspects . . . . .	14
3	SOURCE-AGGREGATED-POISSON . . . . .	19
3.1	The Method . . . . .	19
3.2	Properties of SAP . . . . .	20
	Translation and rotation. . . . .	23
	Scale. . . . .	23
	Pose. . . . .	23
	Occlusions and topology. . . . .	26
3.3	Implementation Details . . . . .	27
4	HIERARCHICAL SHAPE DECOMPOSITION . . . . .	29
4.1	Computation of Preliminary Hierarchies . . . . .	29
4.2	Stochastic Reorganization . . . . .	32
4.3	Comparison with the Decomposition using Previous Nonlocal Model . . . . .	34
5	GROUPWISE ANALYSIS OF A SHAPE SET . . . . .	37
5.1	Method . . . . .	37
5.2	Experimental Results . . . . .	44
6	UNSUPERVISED SEGMENTATION OF 3D MESH MODELS . . . .	49
6.1	Related Work . . . . .	50
6.2	Method . . . . .	52
	Spectral analysis . . . . .	52
	Soft-clustering . . . . .	54

	Hard-segmentation . . . . .	54
6.3	Results and Benchmark Evaluation . . . . .	55
	Limitations . . . . .	60
7	SUMMARY AND CONCLUSION . . . . .	63
	REFERENCES . . . . .	67
APPENDICES		
A	SEGMENTATION PERFORMANCE ON PRINCETON BENCHMARK SET . . . . .	75
B	GROUPWISE ANALYSIS RESULTS WITH 1000-SHAPES DATASET	95
	CURRICULUM VITAE . . . . .	147

## LIST OF TABLES

### TABLES

Table 6.1	Princeton Benchmark Rand Index Values . . . . .	60
-----------	---	----

## LIST OF FIGURES

### FIGURES

Figure 2.1 Mimicking the previous model via <i>repeated</i> application of screened Poisson PDE. . . . .	9
Figure 2.2 The solution $\Phi_r$ to our local model at $r = 0.4$ (first row) versus solution $u$ of the previous nonlocal model (second row). Within each box, from left to right: The functions, their level curves, shape parts implied by deviations from monotonicity, i.e, sign change and saddle points. . . . .	10
Figure 2.3 Solutions $\Phi_r$ at varying values of $r$ (first row) and corresponding segmentations (second row). The associated values of $r$ from left to right are 0.2, 0.25, 0.37, 0.4, 0.56. The midmost example ( $r = 0.37$ ) demonstrates the case where the boundary between the positive and the negative sets just touches the shape boundary. . . . .	10
Figure 2.4 Sample image input: Incomplete snowflake edge map placed in image frames of different sizes, and a crowded image context with junctions as well as missing edges. . . . .	11
Figure 2.5 Sensitivity of the <i>previous model</i> with respect to the size of the image context in which the edge map resides. The zeros of the solutions (the edge map and the sign change locus) are marked in thick black line. When the frame size is increased, the sign change locus in the interior disappears. The six dendrites of the snowflake can no longer be partitioned. . . . .	12

Figure 2.6 Robustness of the <i>proposed model</i> with respect to the size of the image context in which the edge map resides. The zeros of the solutions (the edge map and the sign change locus) are marked in thick black line. Even when the frame size changes, the local behaviour of the level curves remain intact. The sign change locus completes the missing boundary fragments. The six dendrites are similarly partitioned in both situations. . . . .	13
Figure 2.7 Snowflake in a crowded context. The six dendrites are clearly identifiable. The shapes of the central region and of the level curves describing the stellar dendrites are no different than those computed for the snowflake in isolation. . . . .	15
Figure 2.8 Running times comparison of solving the proposed family $\Phi_r$ (chol-solve) and solving function $u$ of the previous model (usolve). See text for the details. . . . .	16
Figure 3.1 SAP computational scheme. Top row: Source functions $f_i$ . Middle row: Corresponding solutions $\Phi_i$ to equation (2.5). Bottom row: Indicator functions $m_i$ for negative sets $\Omega_i^-$ which aggregate into SAP. . . . .	20
Figure 3.2 SAP behavior (left) in comparison with EDT (middle) and screened Poisson (right) fields computed for a 2D deer shape. Corresponding level curves are given in the bottom row. . . . .	22
Figure 3.3 SAP computed for various 2D silhouettes. . . . .	24
Figure 3.4 Response of a solution $\Phi$ to changing scale. Equation (2.5) is solved on 1D lines of different lengths $(x, 2x, 4x)$ with equivalent source functions $f$ . The solutions are mapped onto the line of unit length for comparison at corresponding locations. Observe that the locations of the zero-crossings are preserved through different scales. . . . .	25
Figure 3.5 SAP is robust under changes in the pose of an articulated object. . . . .	25
Figure 3.6 Effect of occlusions and topological changes. . . . .	26

Figure 3.7	SAP computed for the same model with different tessellations (depicted with yellow edges) of the surface and using different grid resolutions (number of voxels).	27
Figure 4.1	Preliminary partitioning hierarchy of an elephant shape (a) and the most probable hierarchy (b) generated by the stochastic reorganization process with probability 0.216.	31
Figure 4.2	Preliminary partitioning hierarchies of a horse shape produced by our method (top) and the method proposed in [68] (bottom).	33
Figure 4.3	The leaf parts of preliminary partitioning hierarchies produced by our method (left) and the method proposed in [68] (right). Observe that our method better localizes the limbs.	34
Figure 5.1	Illustration of the bottom-up processing part of the proposed method for groupwise analysis of a shape set. First, the preliminary hierarchy trees are computed (see Section 4.1). Then, randomized samples are generated using stochastic reorganization (see Section 4.2).	38
Figure 5.2	Illustration of the top-down processing part of the proposed method for groupwise analysis of a shape set. The collective hierarchy is computed as the medoid of the most probable samples of each individual shape. Finally, the optimized individual representations are selected to be the nearest samples to the collective hierarchy.	39
Figure 5.3	The preliminary decomposition hierarchy trees of a paper boat (top) and its wrinkled version (bottom). The comparison shows how boundary deformations can affect the resulting preliminary hierarchies.	41
Figure 5.4	The individual representative hierarchy trees of the paper boats shown in Figure 5.3, i.e. the most probable tree samples generated by the stochastic reorganization process. The comparison shows how boundary deformations can affect the resulting individual representatives.	42

Figure 5.5 The collective decomposition hierarchy tree of the paper boat category (top), and the improved individual decomposition hierarchy of the wrinkled paper boat (bottom), which is the nearest tree to the collective hierarchy tree among all samples generated by the stochastic reordering process for the wrinkled paper boat. Observe also that the first level of the decomposition tree of the wrinkled boat becomes similar to that of the non-wrinkled boat, which was shown in Figure 5.4 to the top, as the further splits of the boat sides are deferred to the next level of the tree. . . . 43

Figure 5.6 The preliminary decomposition hierarchy tree of a cat and elephant silhouette. . . . . 46

Figure 5.7 The collective decomposition hierarchy tree of the cat category (top), the individual representative tree (middle) of the specific cat whose preliminary tree is shown in Figure 5.6, and the optimized individual hierarchy (bottom) obtained by selecting the nearest tree sample to the collective tree. . . . . 47

Figure 5.8 The collective decomposition hierarchy tree of the elephant category (top), the individual representative tree of a specific elephant object (middle), and the optimized individual hierarchy (bottom) obtained for that object. . . . . 48

Figure 6.1 Benchmark performance of our algorithm (Ours) in comparison with state-of-the-art methods of different categories: unsupervised single mesh segmentation algorithms (HGD, [71]), (SSF, [73]) and (M-S, [78]), algorithm for joint segmentation of multiple shapes (Joint, [32]), supervised segmentation with various portions of samples used for training (95%-Sup95, 60%-Sup60, 15%-Sup15, [35]). . . . . 58



Figure 6.2 Evaluation of SAP's contribution to the benchmark performance of our algorithm based on Rand Index values. The contributions are evaluated by using various combinations of the cues in the definition of the distance measure (6.2) as indicated by the labels, i.e. the leftmost bar represents the performance using all cues: SAP function ( $\delta_{SAP}$ ), dihedral angle ( $\delta_{DHD}$ ) and concavity information ( $\kappa$ ). . . . .	59
Figure 6.3 SAP contributes to segmentation performance in different ways. In each triplet, from left to right, we show segmentation results using and not using SAP, and SAP itself. . . . .	59
Figure 6.4 Missing parts due to the current parameter fixations. By adjusting the parameters, we can recover the missing parts. . . . .	61
Figure A.1 Segmentation results for HUMAN category. . . . .	76
Figure A.2 Segmentation results for CUP category. . . . .	77
Figure A.3 Segmentation results for GLASSES category. . . . .	78
Figure A.4 Segmentation results for AIRPLANE category. . . . .	79
Figure A.5 Segmentation results for ANT category. . . . .	80
Figure A.6 Segmentation results for CHAIR category. . . . .	81
Figure A.7 Segmentation results for OCTOPUS category. . . . .	82
Figure A.8 Segmentation results for TABLE category. . . . .	83
Figure A.9 Segmentation results for TEDDY category. . . . .	84
Figure A.10 Segmentation results for HAND category. . . . .	85
Figure A.11 Segmentation results for PLIER category. . . . .	86
Figure A.12 Segmentation results for FISH category. . . . .	87
Figure A.13 Segmentation results for BIRD category. . . . .	88

Figure A.14	Segmentation results for ARMADILLO category. . . . .	89
Figure A.15	Segmentation results for BUST category. . . . .	90
Figure A.16	Segmentation results for MECH category. . . . .	91
Figure A.17	Segmentation results for BEARING category. . . . .	92
Figure A.18	Segmentation results for VASE category. . . . .	93
Figure A.19	Segmentation results for FOURLEG category. . . . .	94
Figure B.1	From decomposition hierarchy tree representation to compact representation with superimposed nodes. The shape itself (root) is drawn with light-gray. The parts that are split in the first level are shown in blue color, whereas the second level is coded using orange. . . . .	96
Figure B.2	Groupwise analysis results for category 1. . . . .	97
Figure B.3	Groupwise analysis results for category 2. . . . .	98
Figure B.4	Groupwise analysis results for category 3. . . . .	99
Figure B.5	Groupwise analysis results for category 4. . . . .	100
Figure B.6	Groupwise analysis results for category 5. . . . .	101
Figure B.7	Groupwise analysis results for category 6. . . . .	102
Figure B.8	Groupwise analysis results for category 7. . . . .	103
Figure B.9	Groupwise analysis results for category 8. . . . .	104
Figure B.10	Groupwise analysis results for category 9. . . . .	105
Figure B.11	Groupwise analysis results for category 10. . . . .	106
Figure B.12	Groupwise analysis results for category 11. . . . .	107
Figure B.13	Groupwise analysis results for category 12. . . . .	108
Figure B.14	Groupwise analysis results for category 13. . . . .	109

Figure B.15 Groupwise analysis results for category 14. . . . .	110
Figure B.16 Groupwise analysis results for category 15. . . . .	111
Figure B.17 Groupwise analysis results for category 16. . . . .	112
Figure B.18 Groupwise analysis results for category 17. . . . .	113
Figure B.19 Groupwise analysis results for category 18. . . . .	114
Figure B.20 Groupwise analysis results for category 19. . . . .	115
Figure B.21 Groupwise analysis results for category 20. . . . .	116
Figure B.22 Groupwise analysis results for category 21. . . . .	117
Figure B.23 Groupwise analysis results for category 22. . . . .	118
Figure B.24 Groupwise analysis results for category 23. . . . .	119
Figure B.25 Groupwise analysis results for category 24. . . . .	120
Figure B.26 Groupwise analysis results for category 25. . . . .	121
Figure B.27 Groupwise analysis results for category 26. . . . .	122
Figure B.28 Groupwise analysis results for category 27. . . . .	123
Figure B.29 Groupwise analysis results for category 28. . . . .	124
Figure B.30 Groupwise analysis results for category 29. . . . .	125
Figure B.31 Groupwise analysis results for category 30. . . . .	126
Figure B.32 Groupwise analysis results for category 31. . . . .	127
Figure B.33 Groupwise analysis results for category 32. . . . .	128
Figure B.34 Groupwise analysis results for category 33. . . . .	129
Figure B.35 Groupwise analysis results for category 34. . . . .	130
Figure B.36 Groupwise analysis results for category 35. . . . .	131

Figure B.37 Groupwise analysis results for category 36. . . . .	132
Figure B.38 Groupwise analysis results for category 37. . . . .	133
Figure B.39 Groupwise analysis results for category 38. . . . .	134
Figure B.40 Groupwise analysis results for category 39. . . . .	135
Figure B.41 Groupwise analysis results for category 40. . . . .	136
Figure B.42 Groupwise analysis results for category 41. . . . .	137
Figure B.43 Groupwise analysis results for category 42. . . . .	138
Figure B.44 Groupwise analysis results for category 43. . . . .	139
Figure B.45 Groupwise analysis results for category 44. . . . .	140
Figure B.46 Groupwise analysis results for category 45. . . . .	141
Figure B.47 Groupwise analysis results for category 46. . . . .	142
Figure B.48 Groupwise analysis results for category 47. . . . .	143
Figure B.49 Groupwise analysis results for category 48. . . . .	144
Figure B.50 Groupwise analysis results for category 49. . . . .	145
Figure B.51 Groupwise analysis results for category 50. . . . .	146

## LIST OF ALGORITHMS

### ALGORITHMS

Algorithm 1	Decomposition of a part . . . . .	30
Algorithm 2	Segmentation for a given number of segments . . . . .	53

## LIST OF ABBREVIATIONS

EDT	Euclidean Distance Transform
PDE	Partial Differential Equation
RI	Rand Index
SAP	Source-Aggregated-Poisson

# **CHAPTER 1**

## **INTRODUCTION**

Shape is generally defined as the form or appearance of an object produced by its outline or external boundary excluding the effects of other visual stimuli such as the object's color, texture and material composition. Throughout this thesis, by using the term shape, we will refer to the volume enclosed by the external boundary of an object. Then, we define shape analysis as processing shapes of objects for the purpose of extracting simplified representations that capture the most significant information and facilitate storage and further processing as required by the application context. In this chapter, we will motivate research on generic shape analysis first from perception point of view and then giving interesting examples of applications.

### **1.1 Shape and Perception**

Vision research has shown that shape plays a critical role in perception. Indeed, we can effortlessly recognize numerous objects just looking at their silhouettes despite different viewing angles and object poses, even when they are partially occluded by other objects.

Neurophysiological findings reveal that encoding of shape information undergoes transformation through different stages of visual pathway from complex and implicit representations at early levels of processing to simple and compact explicit representations that facilitate storage and recognition. Mainly the structural representations are supported where shape parts and their relations are captured. Additionally, there are evidence in favor of some feedback mechanisms that support the bottom-up processing with contextual information and prior knowledge.

## 1.2 Goal and Scope of the Study

In this study, we propose shape analysis techniques to produce representations of shapes for different levels of visual processing that are robust under several visual transformations and occlusions. To this end, we first solve Poisson partial differential equation (PDE) inside the shape with specially designed source functions to explore the shape volume yielding multiple part-coding functions. We show that the proposed model is capable of capturing nonlocal interactions among shape locations, although the computations remain local. Then, we aggregate the rich information provided by those solutions into a single function defined on the shape domain. Using this function, we extract shape decomposition hierarchies as high level structural representations of shapes. Furthermore, we devise a groupwise shape analysis scheme that treats the task of extracting structural representations within the context of similar shapes. The method employs a probabilistic structural reorganization process together with a feedback mechanism to improve representational consistency within the group. We also utilize the function in the context of a mesh segmentation application after projecting it onto the shape boundary.

## 1.3 Motivating Applications of Shape Analysis

The information age and the accompanying digital revolution have made a tremendous amount of visual data available publicly in digital forms such as photos, videos and 3D models as new imaging and manufacturing technologies are entering our lives virtually every day – the cameras on our cell phones, 3D scanners/printers just to name a few. Such a rapid growth in the available data also raises the need for the development of techniques to transform that data into knowledge through which we can understand and change the world for the better.

Extraction of knowledge starts with image understanding where the digital image of a real-world scene is analyzed by discriminating and identifying objects of interest and their relations within the background clutter in order to reveal a high-level semantical interpretation of that scene. At this point, the shape shines out as the most prominent characteristic that can reveal the identity of an object. In order to reveal that potential,



one needs to develop a shape analysis method that yields effective representations of shape.

Understanding part semantics via shape analysis helps robots when interacting with their environment autonomously. For example, it can move an object by recognizing graspable object parts, which requires analyzing object shape and associating functionalities with its parts. Besides this, object tracking is another important function of the robot vision system. In order to track an object, the system should match its shape across sequential frames.

As part of the aforementioned developments, new medical imaging technologies – e.g. magnetic resonance imaging, computed tomography and positron emission tomography scans – revolutionized medicine by enabling visualization and inspection of anatomic structures inside the human body for diagnosis and treatment purposes. Here, shape analysis tools are being developed that are assisting physicians in localizing and tracking of deformations in shapes of anatomic structures for the purpose of inspecting the presence of a disease and its progress.

A big part of the entertainment industry revolves around manipulating shapes. Production of realistic games and impressive animated movies requires the development of effective shape representations accompanied by interpolation and morphing techniques.

Additive manufacturing becomes ubiquitous as the 3D printers get cheaper. However, the technology is relatively new and has its own issues that need solutions. Most of the times, the 3D objects have to be decomposed into parts because of the practical limitations brought by the printing environment. This problem requires the development of automatic shape decomposition tools capable of generating feasible solutions satisfying various constraints related to size, aesthetics, ease of assembly, structural health, etc.

## **1.4 Contributions and Thesis Organization**

This thesis study makes the following contributions:

- A Poisson PDE based local and linear computational model for shape analysis is proposed which can mimic a previous nonlocal model and is even superior in terms of computational efficiency and part-coding capability.
- The rich information gathered by exploring the parameter space of the proposed model is further aggregated into a single function, which we call Source-Aggregated-Poisson (SAP), defined on the shape and can be projected onto its boundary when necessary. SAP is shown to be robust under various visual transformations and nuisance factors including pose changes, occlusions and topological distortions.
- A hierarchical shape decomposition method is proposed that exploits part coding behavior of SAP.
- A groupwise analysis method is proposed for simultaneous decomposition of shape groups with the objective of improving within group representational consistency. To this end, a probabilistic tree reorganization method is employed that enables generation of multiple interpretations of decomposition hierarchies and helps with the selection of optimum individual hierarchies from those interpretations.
- An unsupervised mesh segmentation algorithm is developed that utilizes SAP projected onto the surface mesh. According to benchmark results, the algorithm performs best among unsupervised methods, and even comparable to supervised and groupwise segmentation methods.

In Chapter 2, we define our Poisson PDE based local and linear computational model and compare it to the previous nonlocal model we are inspired by. Then, in Chapter 3, we devise a pooling scheme that further explores the parameter space of the proposed model and aggregates the outcomes into SAP. We describe our hierarchical shape decomposition method together with the stochastic reorganization process in Chapter 4, and further develop a groupwise analysis method in Chapter 5. Finally, we explain our unsupervised mesh segmentation algorithm in Chapter 6 and conclude the thesis in Chapter 7.

## CHAPTER 2

### A LOCAL ALTERNATIVE TO A PREVIOUS NONLOCAL MODEL FOR SHAPE ANALYSIS

In this chapter, we propose a screened Poisson PDE based computationally advantageous local alternative to a previous nonlocal model for shape analysis [66]. Firstly, we review the previous model. We discuss how nonlocal interactions incorporated in their energy minimization formulation result in part-coding behavior. Then, we show how we manage to mimic that part-coding behavior without a global term in the formulation of the energy minimization. Finally, we demonstrate superiority of our local model in terms of the computational efficiency and the flexibility in part-coding capability. In the following chapter, we will extend the new model by exploring the entire shape via multiple biases (modeled as sources and sinks in the Poisson equation), and integrating the outcomes of those explorations into a generic parameter-free tool for volumetric analysis, namely Source-Aggregated-Poisson (SAP).

#### 2.1 Review of the Previous Model [66]

Let  $\Omega \subset \mathbf{R}^n$  be an open set denoting a shape in arbitrary dimensions with boundary  $\partial\Omega$ . In the discrete setting, function  $u$  is computed by solving a *dense* linear system:

$$\left( L - \frac{1}{(\rho_\Omega)^2} I - \beta \frac{1}{|\Omega|} J \right) u = -EDT_\Omega \quad (2.1)$$

where  $L$  denotes the matrix representation of the Laplace operator defined on the shape domain ( $\Omega$ ) subject to homogeneous Dirichlet conditions on the boundary,  $\rho_\Omega$  is the maximal radius,  $I$  is the identity matrix,  $J$  is the matrix of ones (denoting a global summation),  $\beta$  is a scalar, and  $EDT_\Omega$  is the Euclidean distance transform computed within  $\Omega$ .

Roughly speaking, it can be thought as a solution to the following nonlocal energy:

$$\begin{aligned}
& \arg \min_u \int_{\Omega} \left[ |\nabla u|^2 + \frac{1}{(\rho_{\Omega})^2} \left( u(\cdot) - EDT_{\Omega}(\cdot) \right)^2 \right] d\omega \\
& + \beta \int_{\Omega} \frac{1}{|\Omega|} \left( \int_{\Omega} u(\cdot) d\omega \right)^2 d\omega \\
& \text{with } u \Big|_{\partial\Omega} = 0
\end{aligned} \tag{2.2}$$

The last term of the energy involves double integration. The inner integration represents the computation of the global sum and the outer accounts for its contribution to the energy at any shape point.

The energy form makes it explicit that the solution  $u$  is the best approximation of  $EDT_{\Omega}$  subject to two conflicting constraints: being smooth and being oscillatory. In order to be able to stay close to  $EDT_{\Omega}$ , the solution can not be uniformly zero. Hence,  $u$  must take both negative and positive values which cancel each other. This is the only viable way of keeping the third term of the energy (2.2) small. The positive values and negative values tend to cluster due to the imposed regularity by the first term in the energy. The data term in the energy helps too. Positive locus appear in the central part whereas the negative locus in the periphery. This is because the negative values if observed in the central regions (where the value of the  $EDT_{\Omega}$  is higher) cause a higher residual. This causes an increase in the second term of the energy. The balance among the three competing terms gives rise to an emergent sign change locus inside  $\Omega$ . This sign change locus splits  $\Omega$  into coarse central and peripheral structures, and further exploitation of the saddle points enables extracting a hierarchical decomposition of  $\Omega$ .

In the formulation, nonlocal effects are incorporated via two terms, one accounting for boundary based interactions and the other for region based interactions. The region based interaction is modeled as an aggregation over the domain (see the last term in (2.2)). For the boundary based interaction, an indirect approach is adopted. It is incorporated to the model via the second term in (2.2) containing the usual distance transform, as the distance transform codes long distance interactions between pair of boundary points. The choice of these terms in the model is merely for computational reasons; it makes the resulting energy to be minimized a quadratic one [67].

## 2.2 Mimicking the Previous Nonlocal Model Via Local Computations

We envision an oscillatory behavior that mimics part-coding properties of the previous model by computing a suitably regularized approximation to a non-monotone source function  $f : \Omega \rightarrow \mathbf{R}$ . For this purpose, we consider the following regularization problem:

$$\begin{aligned} \arg \min_{\Phi} \int_{\Omega} \left[ |\nabla \Phi|^2 + \alpha (\Phi(\cdot) - f(\cdot))^2 \right] d\omega \\ \text{with } \Phi \Big|_{\partial\Omega} = 0. \end{aligned} \quad (2.3)$$

That is, we induce nonlocality with the help of an external function rather than a nonlocal component, such as  $J$  in (2.1), in the system matrix. The minimizer  $\Phi$  for (2.3) is a smooth approximation of the source function  $f$  where the smoothing level is inversely determined by the positive scalar  $\alpha$ . Its Euler-Lagrange equation is the following screened Poisson PDE:

$$\begin{aligned} (\Delta - \alpha) \Phi &= -\alpha f \\ \text{with } \Phi \Big|_{\partial\Omega} &= 0. \end{aligned} \quad (2.4)$$

That is,  $\Phi$  is a solution to a biased diffusion with the bias being introduced by the scalar  $\alpha$ .

The next question is how to choose the scalar  $\alpha$  and the initial value  $f$ . First, we discuss  $\alpha$ . Rewriting (2.4) as

$$\left( \frac{\Phi - f}{1/\alpha} \right) = \Delta \Phi,$$

$\Phi$  could be interpreted as the solution to the usual linear diffusion equation at time  $T = 1/\alpha$ . Hence,

$$\Phi(\cdot, \frac{1}{\alpha}) = \frac{1}{[2\pi(1/\alpha)]^{n/2}} \exp\left(-\frac{\|\cdot\|_2^2}{2(1/\alpha)}\right) * f$$

That is, the solution is the convolution of the initial function with a Gaussian of width  $\sqrt{\frac{1}{\alpha}}$ , implying that the quantity described by  $\Phi$  diffuses over a range of order  $\sqrt{\frac{1}{\alpha}}$ .

To ensure that the process reaches to the innermost parts of the shape,  $\sqrt{\frac{1}{\alpha}}$  should be on the order of the domain's maximal radius. In order to obtain an absolute behavior, we fix  $\sqrt{\alpha}$  on the order of  $1/\rho_\Omega$ . A simpler decision is to set  $\sqrt{\alpha} = \frac{1}{|\Omega|}$ .

The choice of  $\alpha$  also affects the range of  $\Phi$ . Nevertheless, the range of  $\Phi$  is irrelevant in terms of modeling, thus, can be normalized. To simplify the discussions, we eliminate the effect of  $\alpha$  on the range by scaling the right-hand side. This only scales the range leaving the geometry of the level curves intact, yielding

$$\begin{aligned} (\Delta - \frac{1}{|\Omega|^2}) \Phi &= -f \\ \text{with } \Phi \Big|_{\partial\Omega} &= 0. \end{aligned} \tag{2.5}$$

Next, we discuss selection of  $f$ . The key point is to choose  $f$  such that it is positive inside a putative central structure  $\Omega_c \subset \Omega$ , while being negative on the complement. One possibility is to consider the signed EDT of a central circle as  $f$ . That is the gross structure of the shape is approximated by a circle. Then, the peripheral detail – what remains as the complement of the gross structure – is a deviation from a circle. This approach may have a practical disadvantage: the gross shape is forced to resemble a circle even for elongated, significantly protruded, and bended shapes. Therefore, we adopt a strategy where the shape itself induces some control on the choice of  $f$ . We compute a normalized approximate distance transform of  $\Omega$ ,  $\tilde{d} : \Omega \rightarrow (0, 1]$ , and then compute  $f_r$  as

$$f(x)_r = \begin{cases} 1 & \text{if } \tilde{d}(x) > r \\ -1 & \text{if } \tilde{d}(x) \leq r \end{cases} \tag{2.6}$$

where  $r \in (0, 1]$  is a putative radius and the normalization is performed by dividing the raw distances to their maximum value. Note that, one who is committed to using local and linear computational models may utilize the following approximation [69] of the EDT:

$$\tilde{d} = -\rho \log(1 - v) \tag{2.7}$$

where  $\rho$  is a positive small scalar and  $v$  satisfies the following screened Poisson PDE:

$$\begin{aligned} (\Delta - \frac{1}{\rho^2}) v &= -\frac{1}{\rho^2} \\ \text{with } v \Big|_{\partial\Omega} &= 0. \end{aligned} \tag{2.8}$$

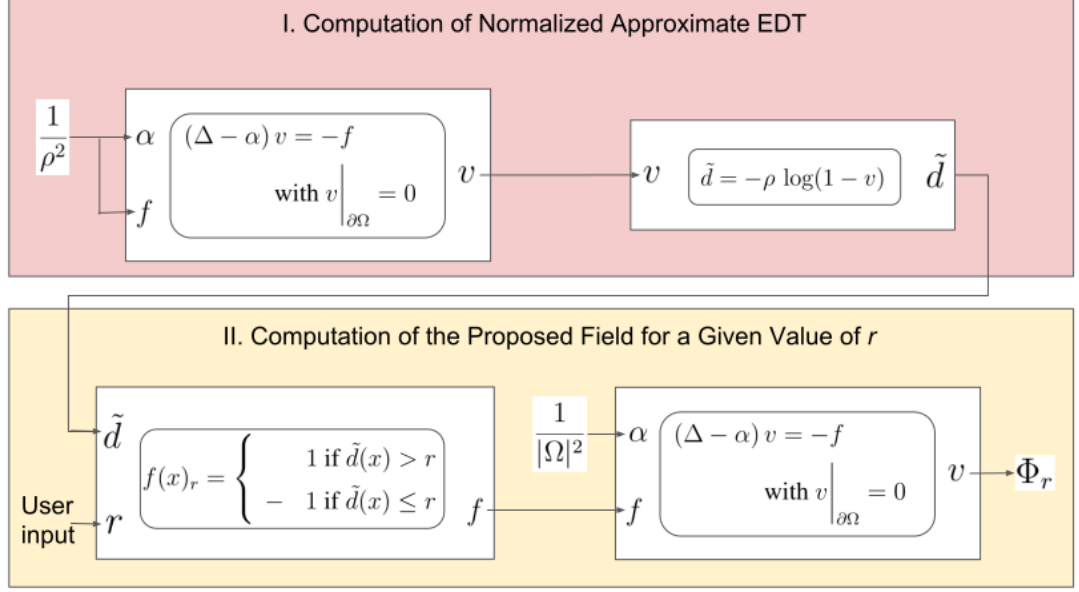


Figure 2.1: Mimicking the previous model via *repeated* application of screened Poisson PDE.

In Figure 2.1, we summarized our two phased computational scheme. In this scheme, screened Poisson PDE is employed in two phases of computation repeatedly; firstly to approximate EDT, and secondly to compute  $\Phi_r$ .

In Figure 2.2,  $\Phi_r$  at  $r = 0.4$  is compared to solution  $u$  of the previous nonlocal model both computed for a snowflake shape. In order to facilitate visual comparison, we separately normalized the positive and the negative set to the unit interval. We also partitioned the shape simply using the watershed algorithm exploiting the locus of sign change and saddle points. Indeed, the two functions and the associated partitionings look quite similar, though not identical. A subtle difference is that the inner level curves of  $\Phi_r$  better reflects the sixfold nature of the snowflake, i.e. early level curves have hexagonal shape rather than being circular.

More examples from the family  $\Phi_r$  are given in Figure 2.3. The figure demonstrates that rather than sticking to a partitioning that result from a single solution, by exploring the whole range of  $r$ , we can obtain partitionings with different granularities.

Then, we can also choose the value of the parameter  $r$  purposefully to capture the solution  $\Phi_r$  whose sign change locus (the boundary that separates the positive and

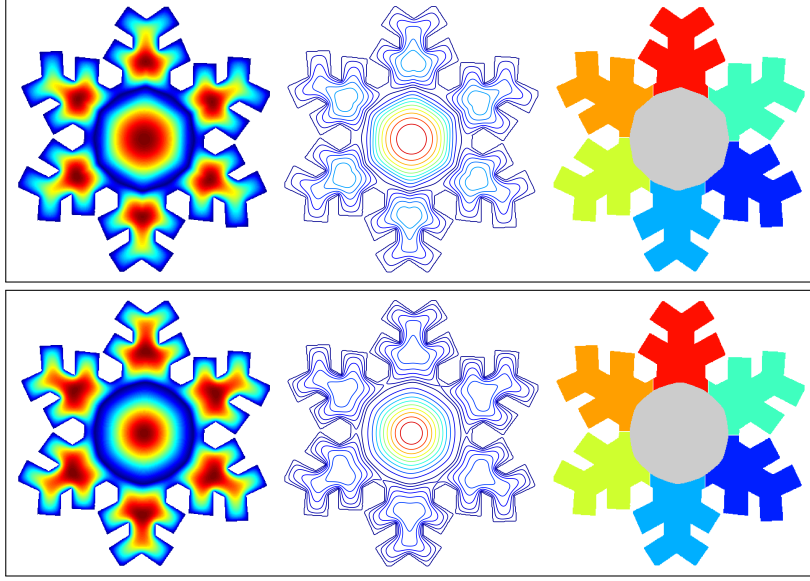


Figure 2.2: The solution  $\Phi_r$  to our local model at  $r = 0.4$  (first row) versus solution  $u$  of the previous nonlocal model (second row). Within each box, from left to right: The functions, their level curves, shape parts implied by deviations from monotonicity, i.e., sign change and saddle points.

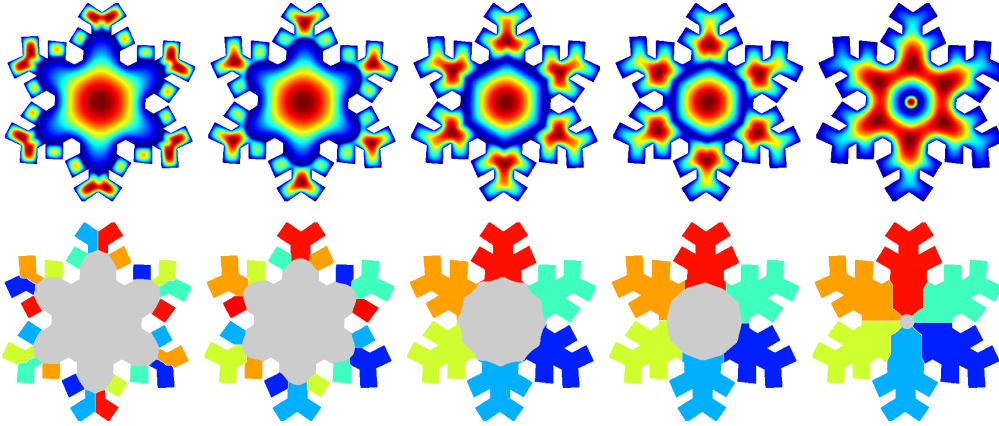


Figure 2.3: Solutions  $\Phi_r$  at varying values of  $r$  (first row) and corresponding segmentations (second row). The associated values of  $r$  from left to right are 0.2, 0.25, 0.37, 0.4, 0.56. The midmost example ( $r = 0.37$ ) demonstrates the case where the boundary between the positive and the negative sets just touches the shape boundary.



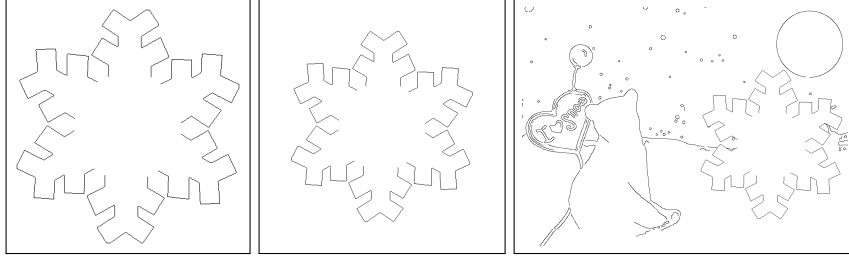
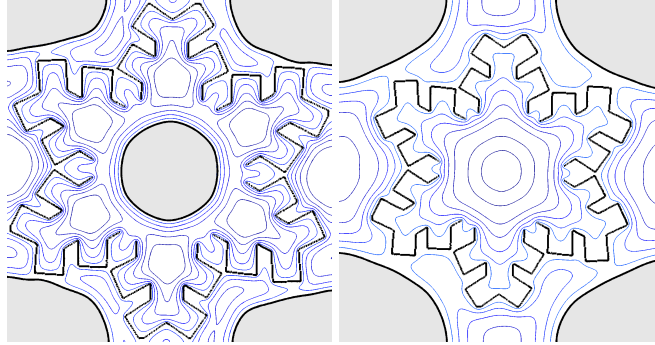


Figure 2.4: Sample image input: Incomplete snowflake edge map placed in image frames of different sizes, and a crowded image context with junctions as well as missing edges.

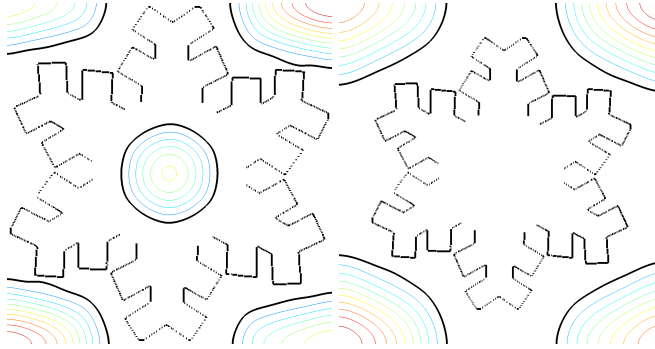
the negative sets) just reaches the shape boundary ( $\partial\Omega$ ) (see the midmost example in Figure 2.3). Our experiments suggest that targeting such a contact between these two contours enables us to effectively find a solution that is analogous to function  $u$  when required. A simple means to spot the right value  $r^*$  that results in such a particular solution is using a binary search like algorithm. Beginning with the solutions at  $r = 0$  and  $r = 1$ , at each step the range between the two selections is halved after checking the solutions  $\Phi_r$ . Then, the iterations are stopped at a point where the range becomes sufficiently narrow. With the precision of  $\epsilon$ , this method finds the target value  $r^*$  in  $\lceil \log_2(1/\epsilon) \rceil$  iterations.

The proposed local model is computationally advantageous, since it reduces into solving a simple sparse linear system unlike the previous nonlocal model, which requires solving a dense linear system. Computational aspects of solving such problems are detailed in Section 2.3.

Another important advantage over the previous nonlocal model emerges in applicability. In an ordinary image application, the shape is first extracted in the form of disconnected edge fragments where the object interior/exterior distinction is not readily available. In order to evaluate the performances of both models, we prepared a simple test input: an incomplete edge map of an isolated snowflake shape that is placed in two image frames with different sizes (see Figure 2.4). Results for the previous model and the proposed alternative are depicted in Figures 2.5 and 2.6, respectively. Sample negative level curves are shown in the top rows, whereas the sample positive

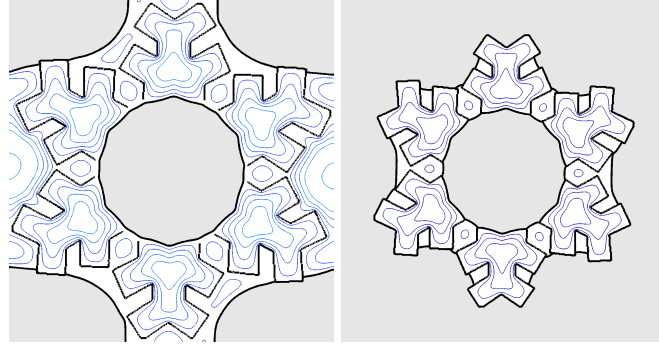


sample negative level curves

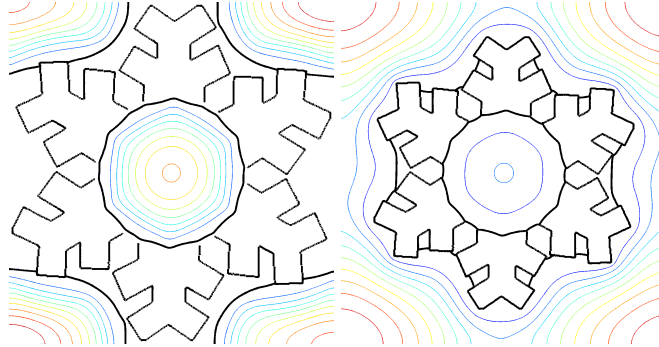


sample positive level curves

Figure 2.5: Sensitivity of the *previous model* with respect to the size of the image context in which the edge map resides. The zeros of the solutions (the edge map and the sign change locus) are marked in thick black line. When the frame size is increased, the sign change locus in the interior disappears. The six dendrites of the snowflake can no longer be partitioned.



sample negative level curves



sample positive level curves

Figure 2.6: Robustness of the *proposed model* with respect to the size of the image context in which the edge map resides. The zeros of the solutions (the edge map and the sign change locus) are marked in thick black line. Even when the frame size changes, the local behaviour of the level curves remain intact. The sign change locus completes the missing boundary fragments. The six dendrites are similarly partitioned in both situations.

level curves in the bottom rows. The zeros (the edge maps and the sign change locus) are marked as thick black lines.

The results show that the previous model is severely sensitive to the size of the image frame in which the edge map resides. The increase in the frame size breaks the non-monotonicity of the solution  $u$  inside the shape, i.e. the distance attains negative values throughout the shape domain and the central structure disappears. This means that, there will not be an interior sign change locus that would have otherwise readily split the shape. Such a behavioral deviation is because of the increase in the number of otherwise irrelevant pixels that contributes to the global average term in (2.1). These irrelevant pixels which are far from the edge map tend to attain positive values pulling the global average towards more positive. Thus, more pixels from inside the shape get negative values in order to level the global average down to zero.

As the sole reason of the sensitivity is the global averaging term, it should not be observed in local models like ours. Indeed, as can be observed in Figure 2.6, the proposed local model is not sensitive to the size of the image context. Moreover, the sign change locus completes the missing boundary sections and the six dendrites are similarly partitioned in both cases.

In Figure 2.7, we go one step further and demonstrate the robustness of our model by embedding the fragmented snowflake shape contours in a crowded image context. Observe that the part of the computed solution covering the snowflake is quite consistent: the level curves describing both the stellar dendrites and the central structure are quite analogous to those computed for the snowflake in isolation (See the top row of Figure 2.2.).

### 2.3 Computational Aspects

We discretize equation (2.5) on a regular grid of voxels (or pixels in 2D) via finite-difference method. Specifically, at each voxel  $v$ , we approximate the Laplacian using the central difference yielding the equation below:

$$-(|\mathcal{N}(v)| + \alpha) \Phi(v) + \sum_{u \in \mathcal{N}(v)} \Phi(u) = -f(v) \quad (2.9)$$

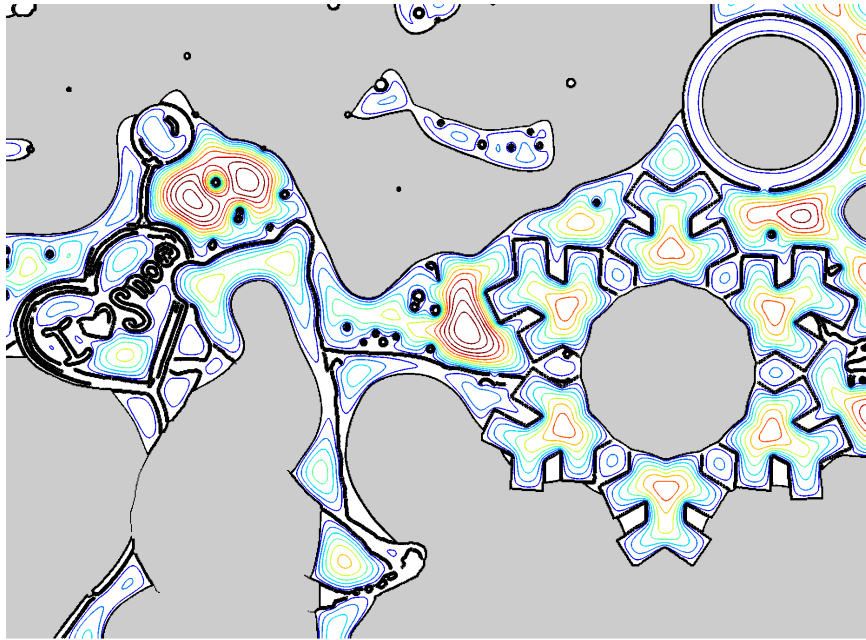


Figure 2.7: Snowflake in a crowded context. The six dendrites are clearly identifiable. The shapes of the central region and of the level curves describing the stellar dendrites are no different than those computed for the snowflake in isolation.

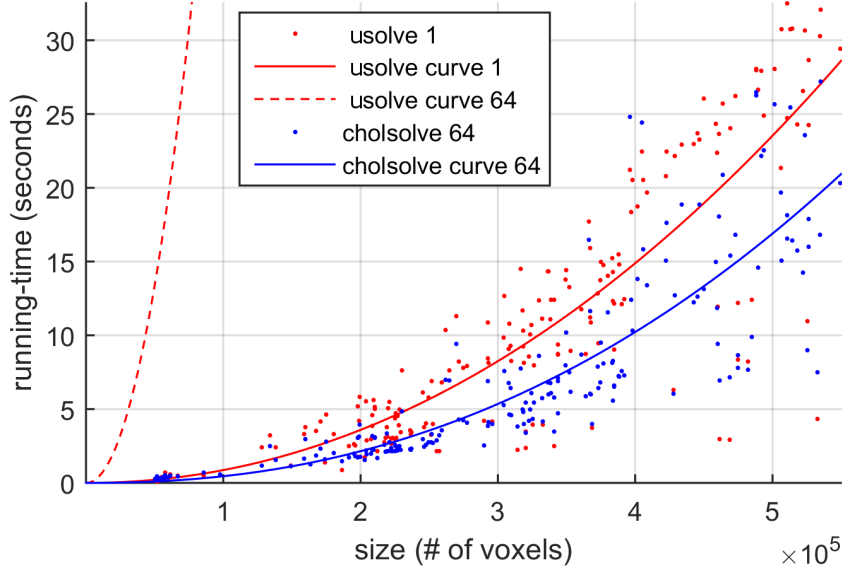


Figure 2.8: Running times comparison of solving the proposed family  $\Phi_r$  (cholsolve) and solving function  $u$  of the previous model (usolve). See text for the details.

where  $\mathcal{N}(v)$  is the set of neighboring voxels and  $|\cdot|$  denotes the cardinality. We used 4- and 6-connected neighboring voxels for 2D and 3D shapes respectively. For the experiments, where shape boundary fragments are put in an image context, we further impose homogeneous Neumann condition on the image boundary.

The discretization yields a system of linear equations with a sparse, symmetric and positive definite coefficient matrix. A plethora of direct and iterative alternatives exist to solve the system with multiple right-hand sides. We use MATLAB built-in CHOLMOD implementation (mldivide function) as a direct solver based on Cholesky factorization. The algorithm is proven to have a sub-quadratic complexity in solving Poisson problem, and performs even better in practice [7]. Once the system matrix is decomposed, the method can solve for large number of right-hand sides very fast. Note here that a fast solution with multiple right-hand sides is crucial both to find the critical value of the parameter  $r^*$  and to extract the rich information via multiple selections of  $r$ .

In Figure 2.8, we compare the running times of the direct CHOLMOD solver (cholsolve) and the iterative solver [68] (usolve) used for obtaining multiple solutions to

our model (2.9) and only a single solution to the previous model (2.1), respectively. Note that constructing a linear system for a multidimensional shape domain in order to compute the function  $u$  of the previous model using a direct solver is not feasible even for moderate grid resolutions and one needs to resort to a relaxation scheme that solves the problem on the grid setting without constructing the system matrix. The solve times were measured for 205 different three-dimensional domains varying in their shape and size ranging from 50k to 550k voxels. When comparing the running times, one should keep in mind that, for each domain, the `usolve` time (red data points) indicates the running time to obtain a single solution for a fixed parameter selection, whereas the `cholsolve` time (blue data points) involves solving (2.9) for 64 different right-hand sides which we found adequate for further processing. In order to facilitate visual comparison, we fit curves to the sample running times. We also put an estimated running time curve (red dashed curve) that corresponds to 64 sequential runs of `usolve` to obtain multiple solutions by varying its parameter  $\rho_\Omega$ . The comparison verifies the computational advantage of our local model over the previous nonlocal model.





## CHAPTER 3

### SOURCE-AGGREGATED-POISSON

In the previous chapter, we developed a Poisson PDE based local and linear computational model for shape analysis that is capable of capturing nonlocal interactions within the shape volume. In this chapter, we will extend the proposed model by further exploring the entire volume via systematically changing its parameter  $r$ , and aggregating the rich information carried in the outcomes of those explorations into a single function, which we call Source-Aggregated-Poisson (SAP).

#### 3.1 The Method

We simultaneously solve  $n$  Poisson equations (2.5) within the shape volume  $\Omega$  by systematically varying the radius  $r$  in (2.6) such that the source function indexed by  $i = 1, 2, \dots, n$  at the  $i$ th step is formulated as

$$f(x)_i = \text{sign}(d(x) - i \times s) \quad (3.1)$$

where  $s = 1/n$  is the step size.

By design, each right-hand side  $f_i$  resembles a heat source in the shape center surrounded by a sink in the outer region which yields a steady state temperature distribution  $\Phi_i$  that divides the shape domain  $\Omega$  into a central region  $\Omega^+$  with positive temperature and a outer region  $\Omega^-$  with negative temperature. Here, the source-sink separation in each right-hand side represents an initial hypothesis for a decomposition of the shape domain into central and marginal regions which correspond to positive and negative sets in the steady state distribution, respectively.

Once a set of  $n$  equations are solved, hence  $n$  distinct solutions  $\Phi_i, i = 1, 2, \dots, n$

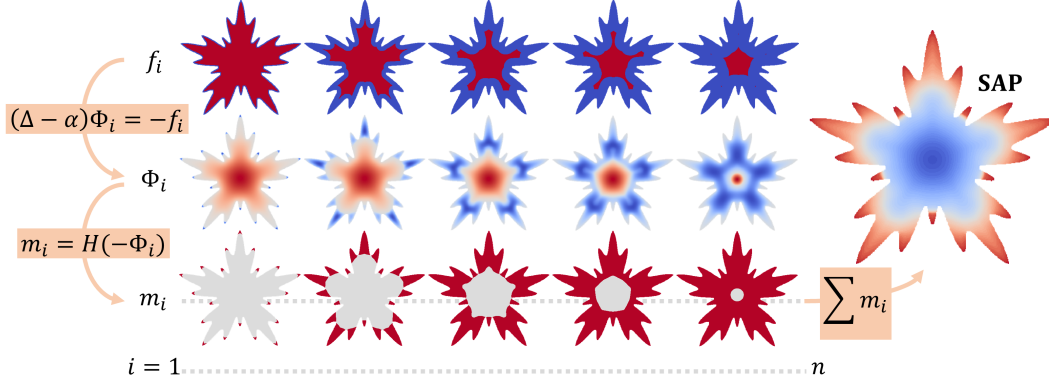


Figure 3.1: SAP computational scheme. Top row: Source functions  $f_i$ . Middle row: Corresponding solutions  $\Phi_i$  to equation (2.5). Bottom row: Indicator functions  $m_i$  for negative sets  $\Omega_i^-$  which aggregate into SAP.

are obtained, the shape information contained in these solutions are aggregated by assigning each shape location the number of solutions in which the location falls into the outer region, i.e. attains a negative value. We call the resulting integer-valued function defined on the shape domain as SAP.

In Figure 3.1, we present an overview of the computational scheme on a two dimensional example. Source functions  $f_i$  are presented in the top row with sources and sinks depicted as red and blue, respectively. Corresponding solutions  $\Phi_i$  to the equation (2.5) are given in the middle row with red tones depicting positive sets  $\Omega_i^+$  and blue tones negative sets  $\Omega_i^-$ . In the bottom row, indicator functions  $m_i$  for the negative sets  $\Omega_i^-$  (Heaviside step function  $H$  applied to the negation of  $\Phi_i$ ) are presented, which are then aggregated to generate SAP.

In order to utilize SAP in applications that require surface features, we further project SAP onto the object surface. To this end, we simply attribute each surface point with the value of the nearest point within the shape domain.

### 3.2 Properties of SAP

The behavior of the SAP is the result of the evolution of the positive set  $\Omega^+$  generated by the sequence of the solutions  $\Phi_i$  ( $i = 1, 2, \dots, n$ ), i.e. moving boundary  $\delta\Omega^+$

form the level curves/surfaces of SAP. Indeed, this is the evolution of the shape itself embedded in the positive sets  $\Omega_i^+$  of the solutions  $\Phi_i$  (with  $\Omega_0^+$  being the original shape  $\Omega$ ), where the inward motion of the boundary  $\delta\Omega^+$  is induced by the systematic change of the source function  $f$  and smoothing is introduced by the operator  $(\Delta - \alpha)$ .

Exploiting the linearity of the operator in (2.5), two consecutive solutions can be related as follows:

$$(\Delta - \alpha)(\Phi_{i+1} - \Phi_i) = -(f_{i+1} - f_i) \quad (3.2)$$

where  $f_{i+1} - f_i$  attains a negative value over the shrinkage zone  $Z_s = \{x : x \in \Omega, i \times s < d(x) \leq (i+1) \times s\}$  and zero otherwise, thus yielding a residual  $\Phi_{i+1} - \Phi_i$  which is negative at any shape location. Negative residual ensures the positive set  $\Omega_{i+1}^+$  to be a subset of  $\Omega_i^+$ . Further taking into consideration the fact that the heat source (positive set of  $f_i$ ) resides in the shape center, we conclude that  $\delta\Omega^+$  evolves towards the center (see  $m_i$  in Figure 3.1). Notice that  $f_{i+1} - f_i$  represents morphological erosion of the shape  $\Omega$  with a radius dictated by the choice of  $s$  and the operator  $(\Delta - \alpha)$  adds a smoothing factor to the evolution process. Thus, together they provide an analogy to the traditional curve evolution where the speed of the evolving curve has a constant (morphological) component and a curvature dependent component for smoothing. However, in contrast to the curve evolution, morphology and smoothing factors act indirectly in producing the behavior of  $\delta\Omega^+$  and hence of SAP as implied by the equation (3.2).

In Figure 3.2, we compare SAP to the Euclidean distance transform  $d(x)$  and to the solution of the screened Poisson equation (2.5) with constant right-hand side  $f = f(x)_0 = 1$  over a 2D example. Here, our goal is to discuss the behavior of SAP with respect to those fields and to emphasize the difference of SAP from a single solution to the screened Poisson, where the latter represents a sample constituent of the former aggregate construction. The level curves of the distance transform simulates curve evolution with constant speed (pure morphological evolution) and those of the screened Poisson solution field represent curvature dependent motion [70]. The level curves of the SAP reveals the effect of morphological and smoothing factors on the evolution of  $\delta\Omega^+$  as well as the distinguishing property that it can remain stationary for a certain time, i.e. the contours of SAP, except the innermost ones, partly coincide

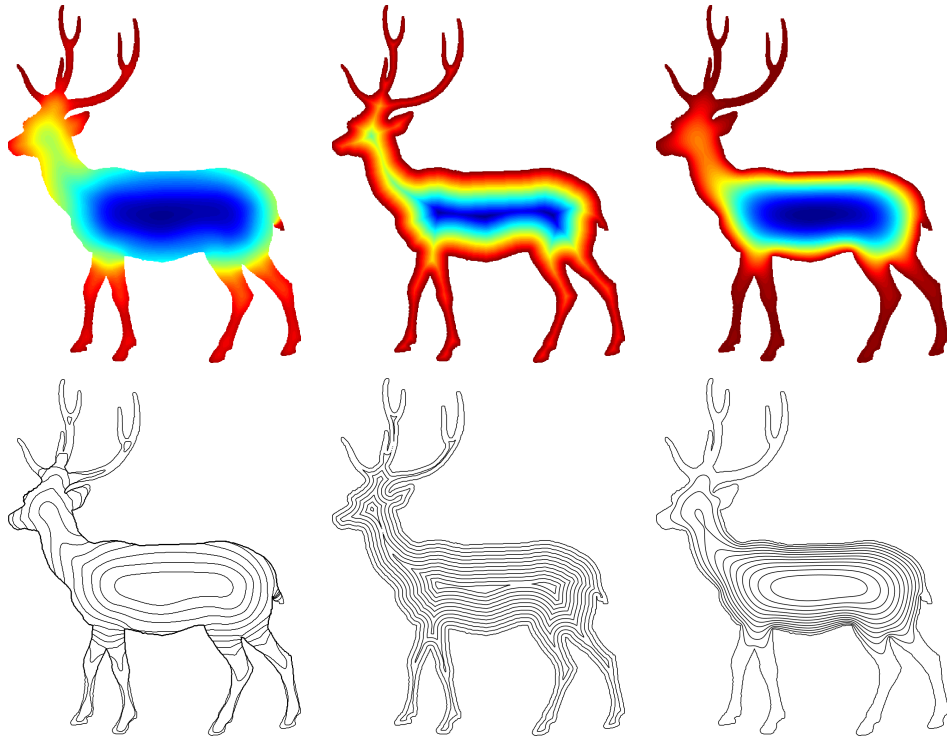


Figure 3.2: SAP behavior (left) in comparison with EDT (middle) and screened Poisson (right) fields computed for a 2D deer shape. Corresponding level curves are given in the bottom row.

with the boundary of the shape.

In Figure 3.3, we demonstrate the behaviour of SAP for further 2D silhouettes. Regions that the positive set  $\Omega^+$  leaves early are depicted with red tones, whereas blue tones represent central regions held by  $\Omega^+$  till the last stages of the evolution. Observe that the inward motion of  $\delta\Omega^+$  starts near locally narrow regions such as the tips of protrusions, corners and narrow necks. During the course of the evolution, those details are smoothed out and  $\delta\Omega^+$  becomes gradually coarser versions of the shape boundary until the positive set disappears.

In the following paragraphs, we investigate the behavior of the SAP under several transformations and nuisance factors such as pose changes, occlusions and topological differences.

**Translation and rotation.** A solution  $\Phi_i(x)$  at any shape point  $x \in \Omega$  to the equation (2.5) with homogeneous boundary conditions depends on the location of  $x$  with respect to the shape boundary  $\delta\Omega$  up to the choice of the source function  $f_i$  which is also derived from a translation and rotation invariant feature, namely Euclidean distance transform. SAP inherits translation and rotation invariance from its constituents  $\Phi_i$ .

**Scale.** We identify the scale with the volume of the shape. Change of the volume scales the range of the solutions  $\Phi$  to the equation (2.5) leaving the locations of sign change (or  $\delta\Omega^+$ ) and hence the behavior of SAP unchanged. Note that the source function has already been made scale invariant using normalized distance transform. In Figure 3.4, we demonstrate corresponding solutions  $\Phi$  for 1D lines of different lengths ( $x, 2x, 4x$ ). Here, different lengths imply different number of elements used for discretization, which are assumed to represent the same unit length. For visual comparison of the solutions at corresponding locations, we mapped each of them onto a line of unit length.

**Pose.** We demonstrate the robustness of SAP under pose changes of articulated models, e.g. moving mammals, which are near-isometric deformations of shapes.

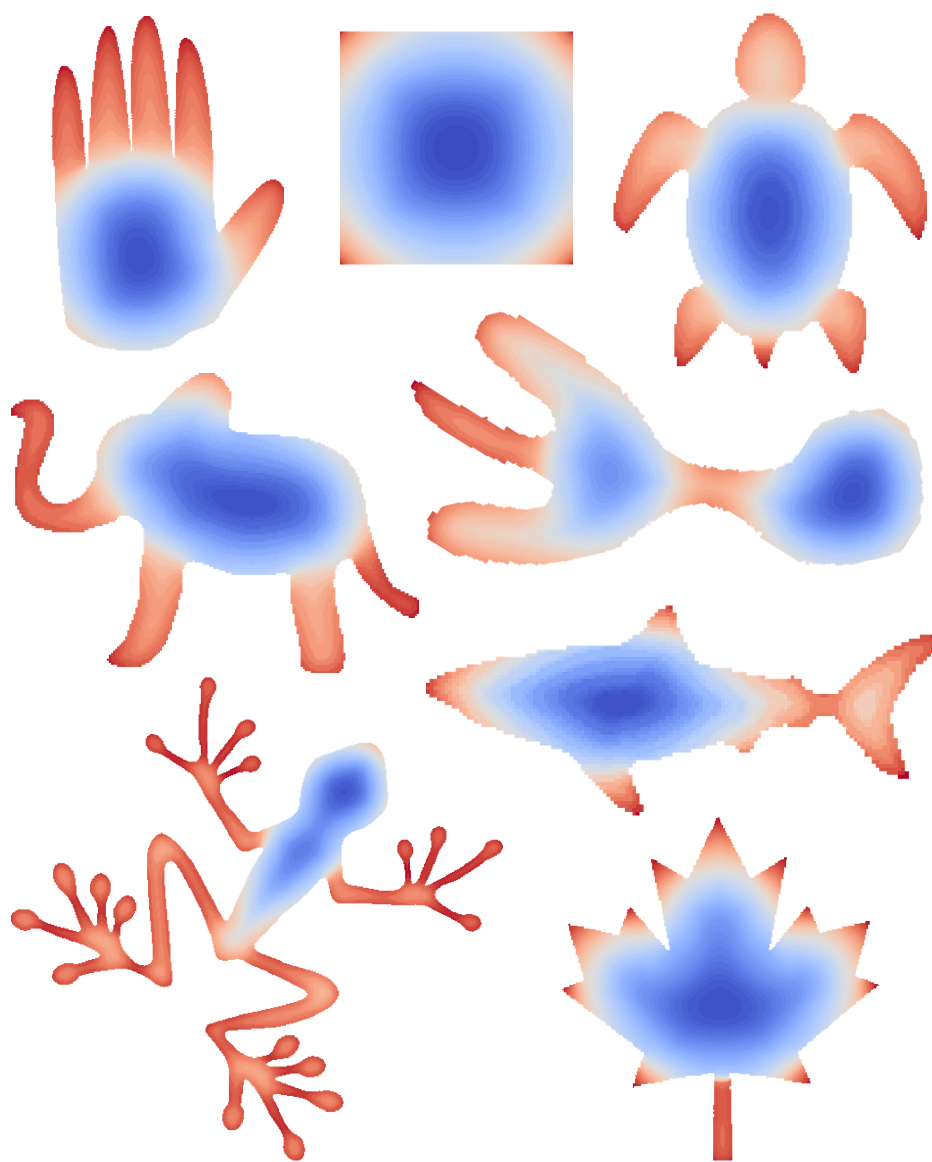


Figure 3.3: SAP computed for various 2D silhouettes.

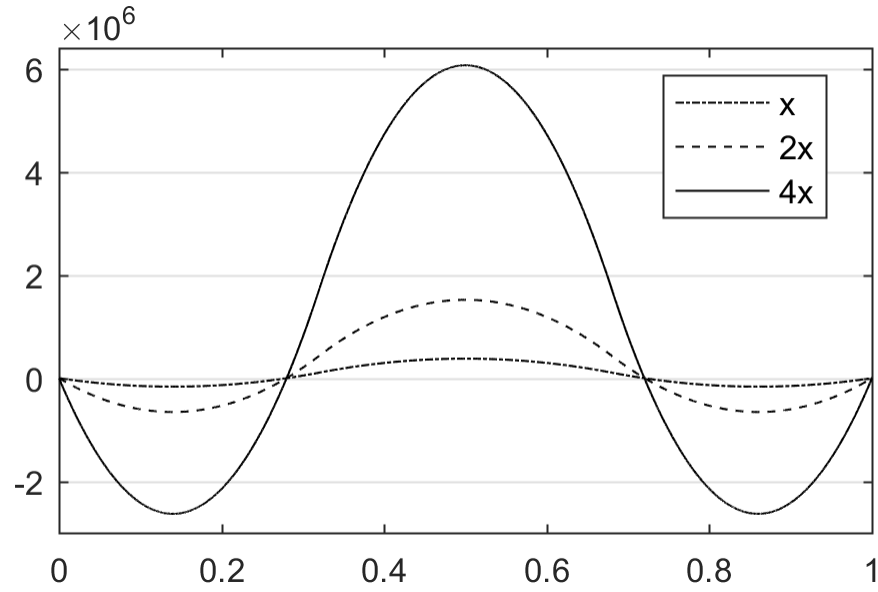


Figure 3.4: Response of a solution  $\Phi$  to changing scale. Equation (2.5) is solved on 1D lines of different lengths ( $x$ ,  $2x$ ,  $4x$ ) with equivalent source functions  $f$ . The solutions are mapped onto the line of unit length for comparison at corresponding locations. Observe that the locations of the zero-crossings are preserved through different scales.



Figure 3.5: SAP is robust under changes in the pose of an articulated object.

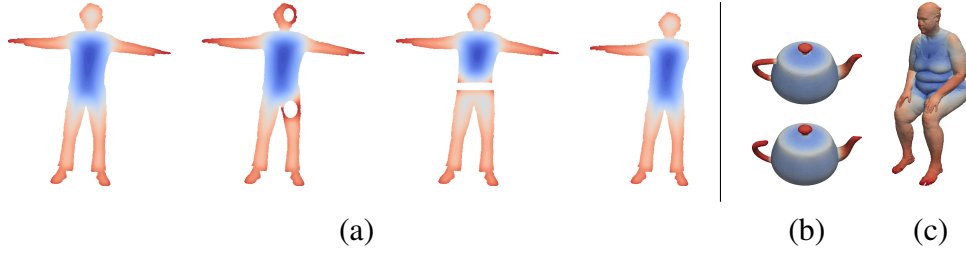


Figure 3.6: Effect of occlusions and topological changes.

Toward this end, we experiment with different poses of a human body from scape dataset [1]. The dataset consists of 71 registered mesh models. In Figure 3.5, we illustrate SAP computed for various representative poses. We also compute the average deviation to report a statistical measure. We first calculate the mean absolute deviation of SAP at each surface element across 71 poses and then average this value over all surface elements. The average deviation is measured about 1% of the specified resolution  $n$  used for determining the step size  $s$ . Several factors are responsible of the measured deviation including anatomically-based deformations of the body volume, noise in the mesh data due to acquisition of 3D point data, surface reconstruction and registration errors, and discretization of our computational model.

**Occlusions and topology.** In real life scenes, we often encounter objects partially occluded by other objects, which may also affect the topology of the object’s shape. Our experiments show that the effect of the occlusion is fair and localized to the vicinity of the occlusion (see Figure 3.6a) with the exception that the central blob is occluded to a large extent. SAP is also resilient to modifications in part connectivity such as the modification of the mounting style of the handle to the teapot shown in Figure 3.6b. In an extreme case, an articulation might entirely merge with the body as shown in the example of the woman in Figure 3.6c whose arms are entirely sticks to the body and legs. The effect on SAP appears still limited.



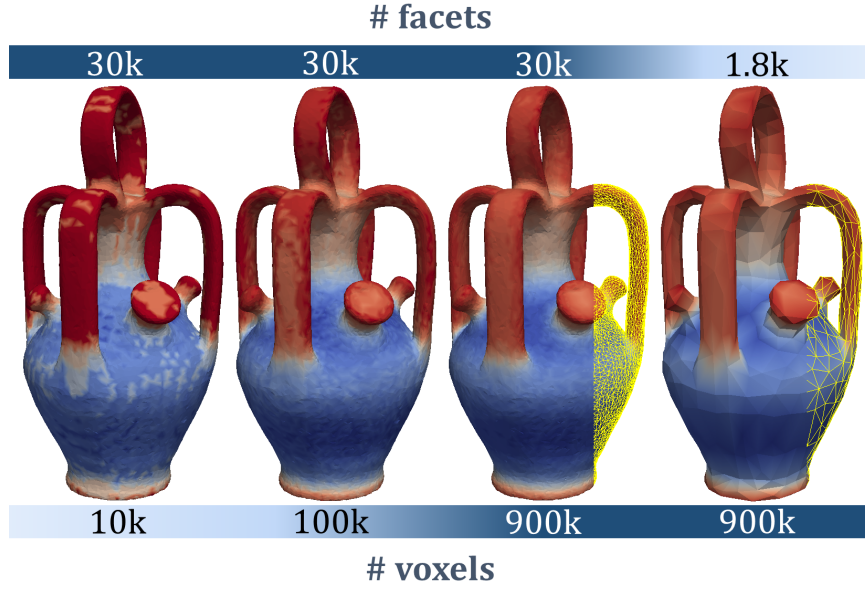


Figure 3.7: SAP computed for the same model with different tessellations (depicted with yellow edges) of the surface and using different grid resolutions (number of voxels).

### 3.3 Implementation Details

A proper grid resolution should be chosen considering the task dependent speed and quality requirements. Quality of SAP depends mostly on the grid resolution specified for voxelization rather than the surface tessellation. In Figure 3.7, we demonstrate the effect of changing the grid resolution (number of voxels) and the tessellation (number of facets) using a vase model. SAP appears to be resistant to a  $16\times$  coarser tessellation and behaves well when the vase volume is represented using 100k voxels or more, whereas noise becomes apparent using 10k voxels. Averaging over a dataset of 380 3D shapes, the computation of SAP for a shape with 100k, 500k and 900k voxels took approximately 1.3, 20 and 77 seconds respectively on a desktop computer with 4 GHz quad-core i7 CPU. We also note that the specified step size  $s$  (or the number of distinct solutions  $n$ ) which determines the resolution of the SAP imposes a minimum grid resolution such that the maximum distance of a point in the shape, in terms of the number of voxels, should at least be on the order of  $n$ .



## CHAPTER 4

### HIERARCHICAL SHAPE DECOMPOSITION

We evaluate the effectiveness of SAP for use in a hierarchical shape decomposition task. To this end, we devised a simple method that exploits evolutionary behavior of the level curves of SAP to extract a decomposition hierarchy of a shape. The method is detailed in Section 4.1. We will call the output tree the preliminary partitioning hierarchy tree which we consider an intermediate representation that is then exposed to a stochastic reorganization process in order to obtain a more intuitive hierarchy and to improve stability among the hierarchies of similar shapes enabling multiple shape interpretations. The reorganization process performs local rearrangements on the hierarchy tree based on the difference between SAP levels of partitioning as explained in Section 4.2.

#### 4.1 Computation of Preliminary Hierarchies

Let the root node hold the shape and each child node represent a sub-part of its parent node in the preliminary partitioning hierarchy tree. We apply a simple thresholding based procedure to split a part into its subparts (see Algorithm 1).

Two components of the decomposition algorithm are connected component analysis of the binarized part volume based on the SAP level threshold and pruning spurious components. The pruning step consists of size and protrusiveness tests. A candidate component is considered too small if its volume is not larger than 0.5% of the shape volume. The degree of a part's protrusiveness measure, as suggested by Hoffman and Singh [30], is the ratio of the part boundary length to the length of the base line that connects the two boundary end points of the part. In order for a candidate part to

---

**Algorithm 1:** Decomposition of a part

---

**input :**  $SAP_{\mathcal{P}}$ , where  $\mathcal{P}$  denotes the part domain.

Degree of protrusiveness threshold  $dopth$ .

Part size threshold  $sth$ .

**output:** Set of subparts  $C$ .

**begin**

$th \leftarrow \min(SAP_{\mathcal{P}});$

$split \leftarrow false;$

**while**  $th < \max(SAP_{\mathcal{P}})$  **and not**  $split$  **do**

$C \leftarrow \text{ConnectedComponentAnalysis}(SAP_{\mathcal{P}} > th);$

$C \leftarrow \text{Pruning}(C, sth, dopth);$

**if**  $|C| > 1$  **then**

$split \leftarrow true;$

**else**

$C \leftarrow \emptyset;$

**end**

$th \leftarrow th + 1;$

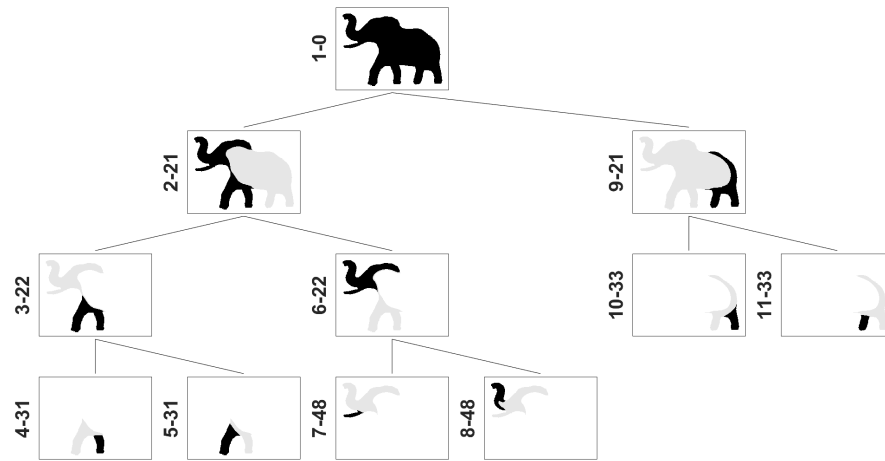
**end**

**end**

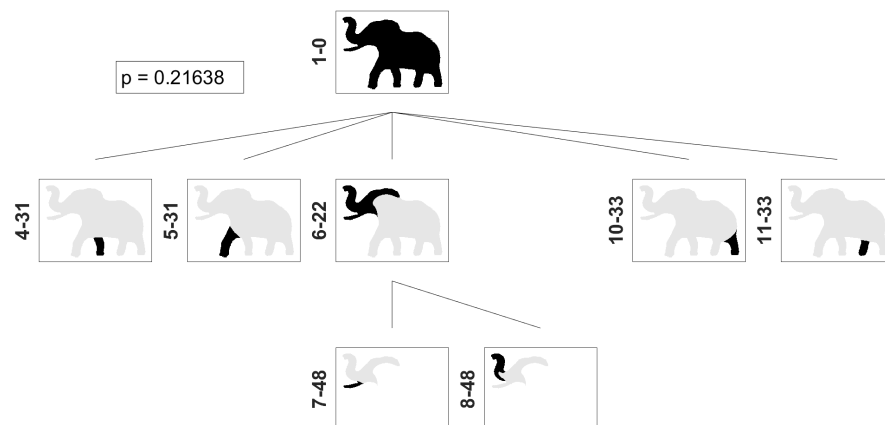
---

qualify the protrusiveness test, it has to meet two criteria. First, the base line should intersect the shape volume indicating the candidate is actually a protrusion. Second, the measured degree of protrusion ratio should at least be 1.3. Note here that all threshold values are determined experimentally.

In Figure 4.1 (a), the preliminary partitioning hierarchy tree of an elephant shape is demonstrated. At each tree node, the corresponding part is depicted black, and the residual area from the parent part is shaded for visual comparison. Note that the union of sibling parts does not cover their parent completely. Numeric code pairs (ID-Split value) next to tree nodes indicate part identification numbers followed by SAP levels at which the parts are split.



(a)



(b)

Figure 4.1: Preliminary partitioning hierarchy of an elephant shape (a) and the most probable hierarchy (b) generated by the stochastic reorganization process with probability 0.216.

## 4.2 Stochastic Reorganization

The naive decomposition approach is prone to yield trees that contain inconsistencies in the hierarchy levels of parts perceived similar. For example, in Figure 4.1 (a), forelegs and hind legs of the elephant appear at different levels of the hierarchy. The reason is the imbalance between the siblings on the second level in the hierarchy. The sibling (ID:2) that holds the front portion of the elephant includes additionally the head, the trunk and the tusk parts, which in turn postpones the split of the legs.

We propose a stochastic reorganization method in order to address the consistency issue and to enable a probabilistic way of shape representation that could help select the task-dependent optimum from among multiple possible hierarchies.

At the core of the reorganization process is the random decision made locally at each internal node (except the root) of whether to replace the node with its children, i.e. the children become the children of their ancestor. The process relies solely on the difference between the split (SAP) value of the parent and children in determining the probability of a potential replacement decision. Specifically, the probability of replacement  $p_r$  is calculated using the formula  $\exp(-\lambda(l_c - l_p))$ , where  $\lambda = 0.05$  determines the rate of decay, and  $l_c$  and  $l_p$  are the split values of children and parent nodes, respectively. In order to reserve some space for randomness at both extremities of the split level difference, i.e. when  $l_c - l_p$  becomes small or large, we impose a maximum (0.85) and minimum (0.05) for  $p_r$ .

In Figure 4.1 (b), the most probable hierarchy tree is shown that is generated by the reorganization process described above for the same elephant shape whose preliminary partitioning hierarchy was shown in part (a) of the same figure. The process generates this hierarchy with probability 0.216. Observe the nodes 2, 3 and 9 are replaced by their children. The reorganization process improves the hierarchy in two ways. Firstly, conforming to our expectation, the legs (4, 5, 10, 11) share the same level in the hierarchy together with the head as a whole, from which then the tusk (7) and the trunk (8) split off. Secondly, the process tends to remove parts that represent perceptually less significant abstractions such as front (2), back (9) and forelegs (3), yielding a more compact partitioning hierarchy.

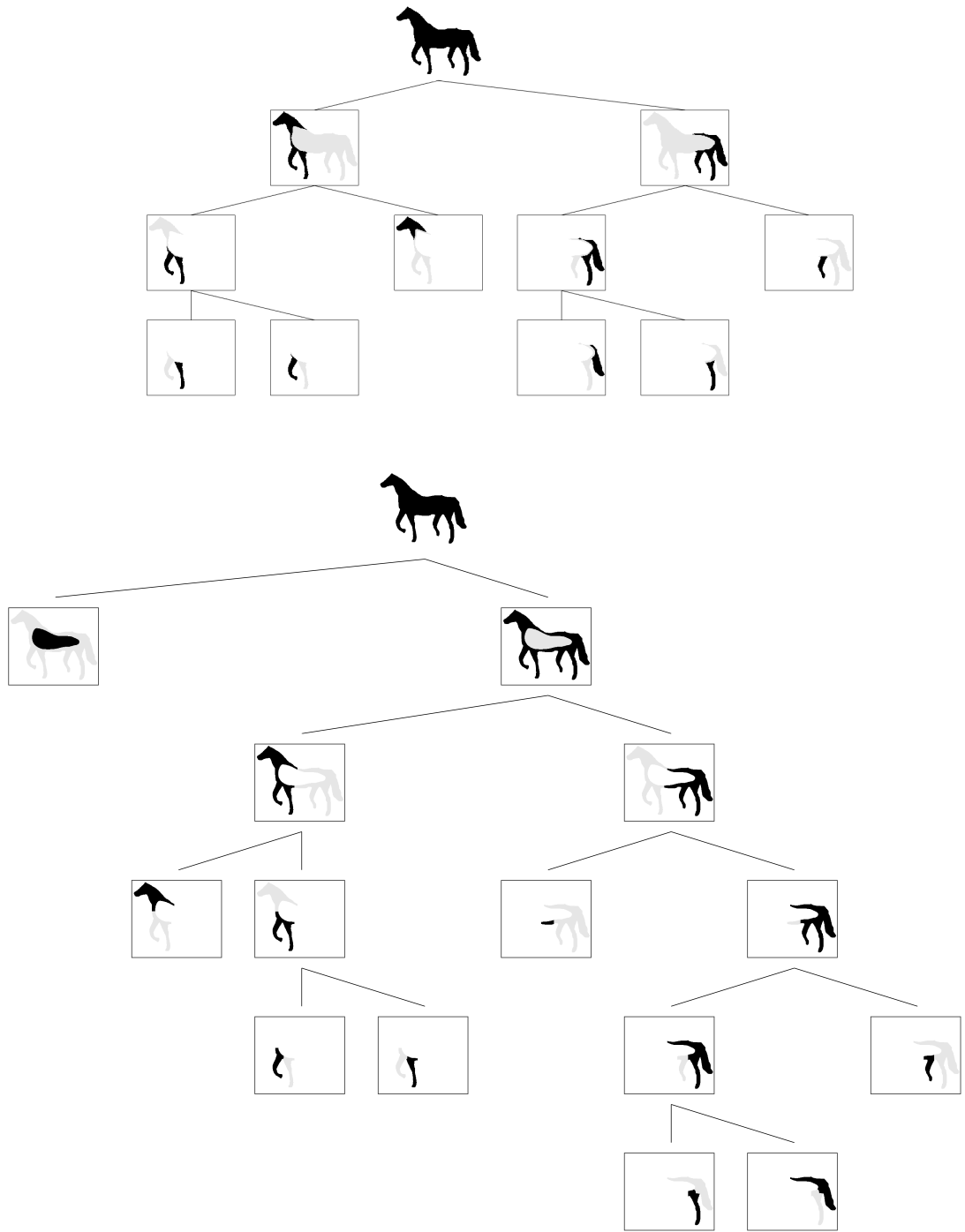


Figure 4.2: Preliminary partitioning hierarchies of a horse shape produced by our method (top) and the method proposed in [68] (bottom).



Figure 4.3: The leaf parts of preliminary partitioning hierarchies produced by our method (left) and the method proposed in [68] (right). Observe that our method better localizes the limbs.

### 4.3 Comparison with the Decomposition using Previous Nonlocal Model

In [68], a method of hierarchical shape decomposition has been proposed that exploits the geometry and topology of the level curves of solution  $u$  to the previous nonlocal model which was reviewed in Chapter 2. The method decomposes the shape domain into adjacent regions enclosed by the level curves of  $u$  that are passing through saddle points of the function. Repeated such splits ordered by the levels of the saddle points yield a decomposition hierarchy that can be represented by a hierarchy tree.

The most important difference is that our method of decomposition is easier to implement or compute than the method proposed in [68]. Remember that our method just thresholds SAP values and the latter needs to locate saddle points of  $u$  which, in turn, requires computation of derivatives up to second degree which especially limits extendability to higher dimensions.

In Figure 4.2, we compare the methods over the decomposition of a horse shape. A significant difference is that the decomposition method in [68] first extracts the coarse central blob (see the left sibling of the first level of decomposition hierarchy) that is the positive locus of  $u$  (see Section 2.1). The saddle point based decomposition of the peripheral parts starts after that point. Our SAP based method does not extract such a central region explicitly. However, the difference set between the whole shape and the parts extracted at the first level of decomposition can be considered a central structure as well.



A more subtle difference is observed in the coverage relationship between the parent-child nodes. Combination of the sibling parts that result from  $u$  based decomposition method reproduces the parent part, whereas the combination of the siblings do not reproduce the parent in our method. We consider this a useful property of our method which provides the flexibility for better localization of parts. Observe that the legs, which are shown in the leaves of the hierarchy tree, are better localized in comparison to those produced by the other method that excessively penetrate into the center of the shape. We depicted the leaves of both decomposition trees in Figure 4.3 for an easier visual comparison.



## CHAPTER 5

### GROUPWISE ANALYSIS OF A SHAPE SET

The task at hand may require analysis of a group of similar shapes simultaneously, where it is suitable to gather information from individual shape analyses to obtain the collective partitioning hierarchy representing the group. Then, the collective hierarchy could help optimizing the individual hierarchies in an attempt to reduce within group representational variance by selecting the hierarchies that are most similar to the collective hierarchy from the set of all possible sample hierarchies that can be generated by the stochastic reorganization process.

#### 5.1 Method

The overview of our method is depicted in Figures 5.1 and 5.2 separating bottom-up and top-down processing parts. In the bottom-up part, we employ the hierarchical partitioning and stochastic reorganization methods given in Chapter 4 to obtain sample decomposition hierarchies.

In the top-down part, the collective hierarchy is determined as the medoid of the set of the individual representatives. The individual representatives are the most probable hierarchy trees generated by the stochastic reorganization process for each individual. In computing the medoid we utilize the tree editing distance algorithm proposed in [79] to measure pairwise distances between individual representatives. Once the pairwise distances are computed, the individual representative whose median distance to other individual representatives is minimum qualifies as the collective hierarchy of the group. Note that we use the median instead of the average distance to prevent the outliers to influence the process adversely.

## Bottom-up Processing

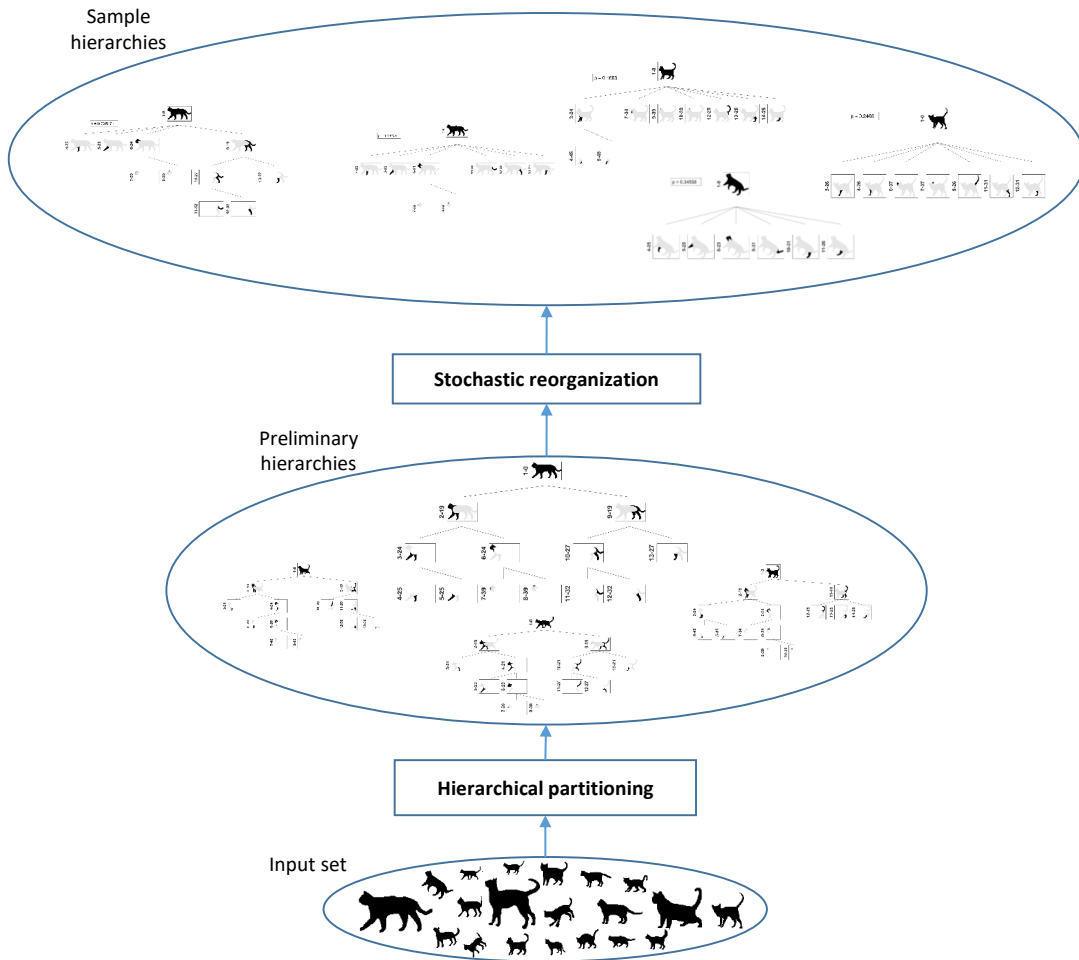


Figure 5.1: Illustration of the bottom-up processing part of the proposed method for groupwise analysis of a shape set. First, the preliminary hierarchy trees are computed (see Section 4.1). Then, randomized samples are generated using stochastic reorganization (see Section 4.2).

## Top-down Processing

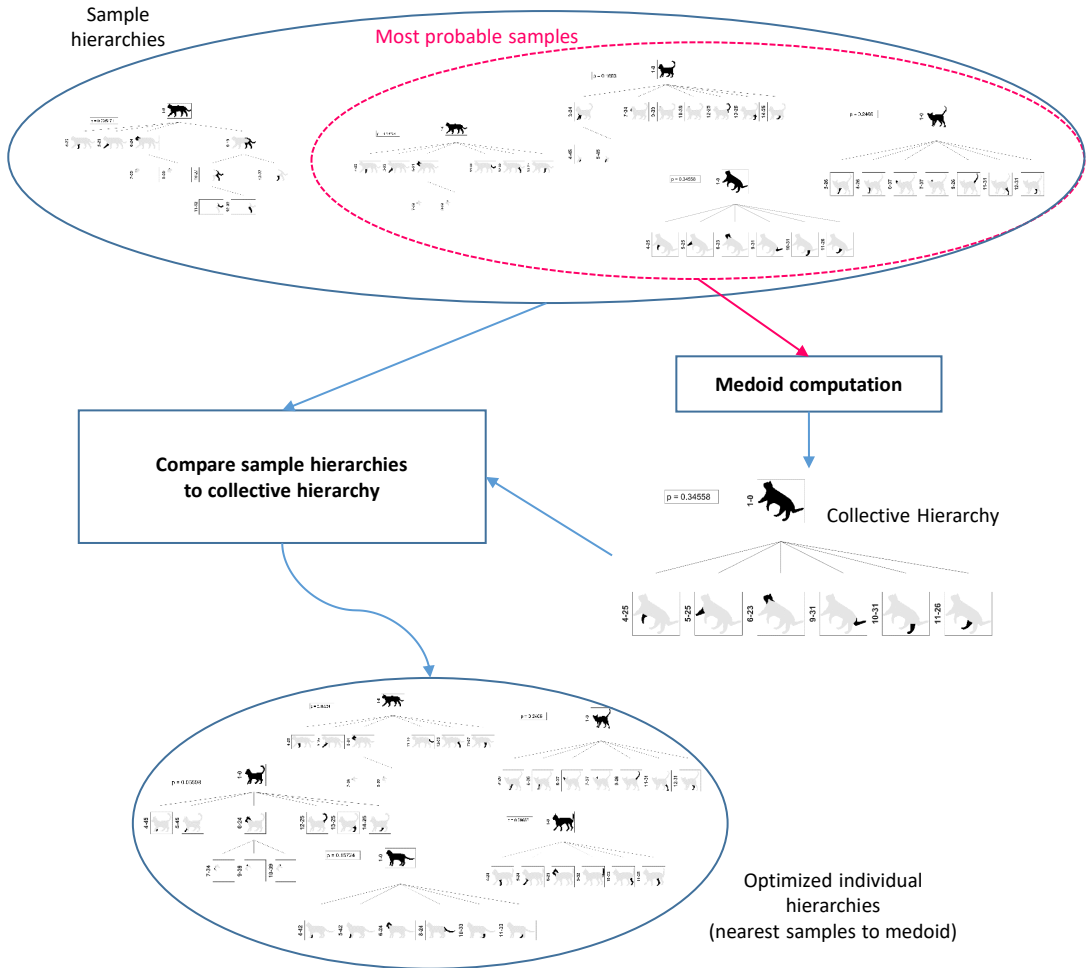


Figure 5.2: Illustration of the top-down processing part of the proposed method for groupwise analysis of a shape set. The collective hierarchy is computed as the medoid of the most probable samples of each individual shape. Finally, the optimized individual representations are selected to be the nearest samples to the collective hierarchy.

The tree editing distance algorithm was proposed for ordered labeled trees, i.e. the order among the siblings is important and the nodes are assigned labels from a set of symbols. Thus, we have to address two issues to be able to employ the algorithm in our method: ordering the sibling nodes and proposing a shape characteristic as a replacement for label abstractions. The proposed characteristic, then, allows us to specify the costs of tree editing operations which are needed to transform a hierarchy tree to another as a means of measuring the distance between those trees.

We order the siblings with respect to their clockwise order which is captured by tracing the shape boundary. We use the part to entire shape area ratio, namely the relative area of a part, as the shape characteristic property of tree nodes to determine the cost of an editing operation. Specifically, the cost is equal to the absolute difference between the relative area properties of two nodes in case of a change operation, i.e. the two nodes are matched. Otherwise, when a node is not mapped to any node in the other tree, i.e. it is deleted or inserted, the node contributes as much as its relative area to the overall tree editing distance.

Finally, for each individual shape, we select the most similar sample hierarchy to the collective one among all samples generated by the stochastic reordering process. Again, we use the tree editing distance algorithm with a small modification in defining the costs of individual operations. Specifically, an insertion operation adds half the cost of a deletion operation.

The rationale behind the weighting idea is the assumed precedence of the collective hierarchy over individual hierarchies. Thus, we evaluate the absence of a collective hierarchy node in the individual hierarchy as a more significant aberration than the presence of a node in the individual hierarchy that has no counterpart in the collective hierarchy. This assumption provides us the necessary degree of flexibility to select individual hierarchy representations that capture as much of the collective information as possible while preserving the variation in the individual hierarchies.

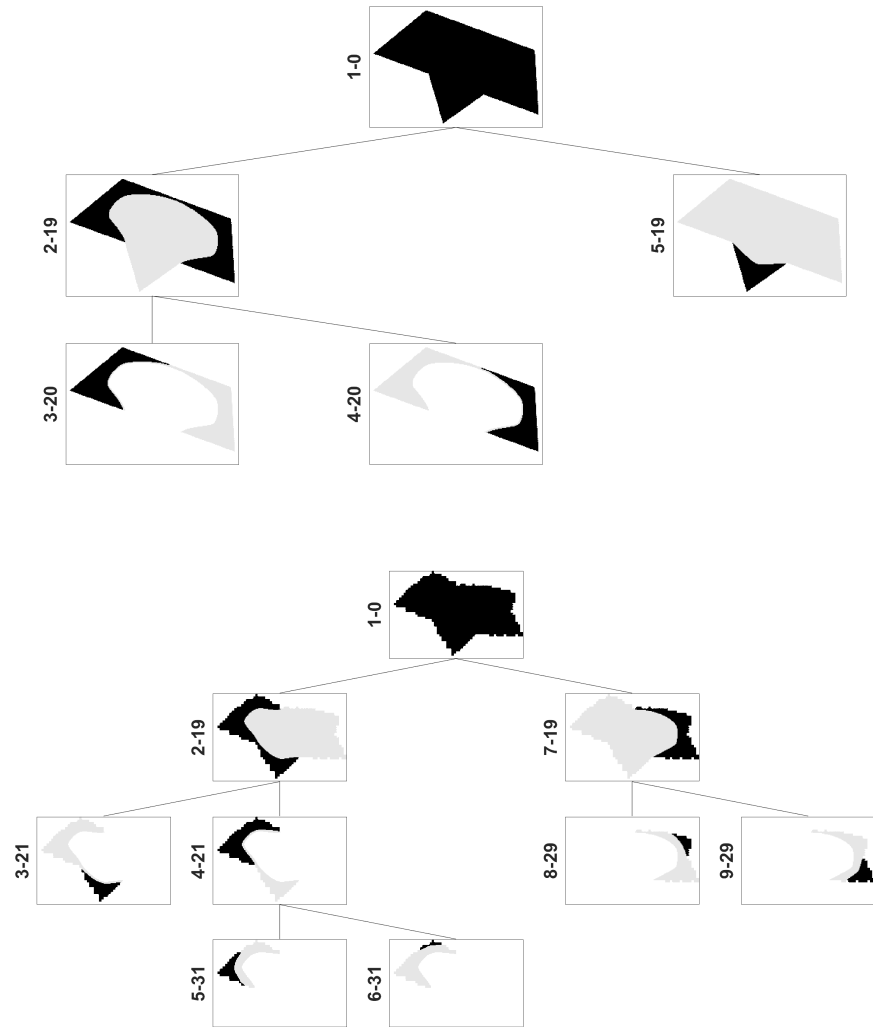


Figure 5.3: The preliminary decomposition hierarchy trees of a paper boat (top) and its wrinkled version (bottom). The comparison shows how boundary deformations can affect the resulting preliminary hierarchies.

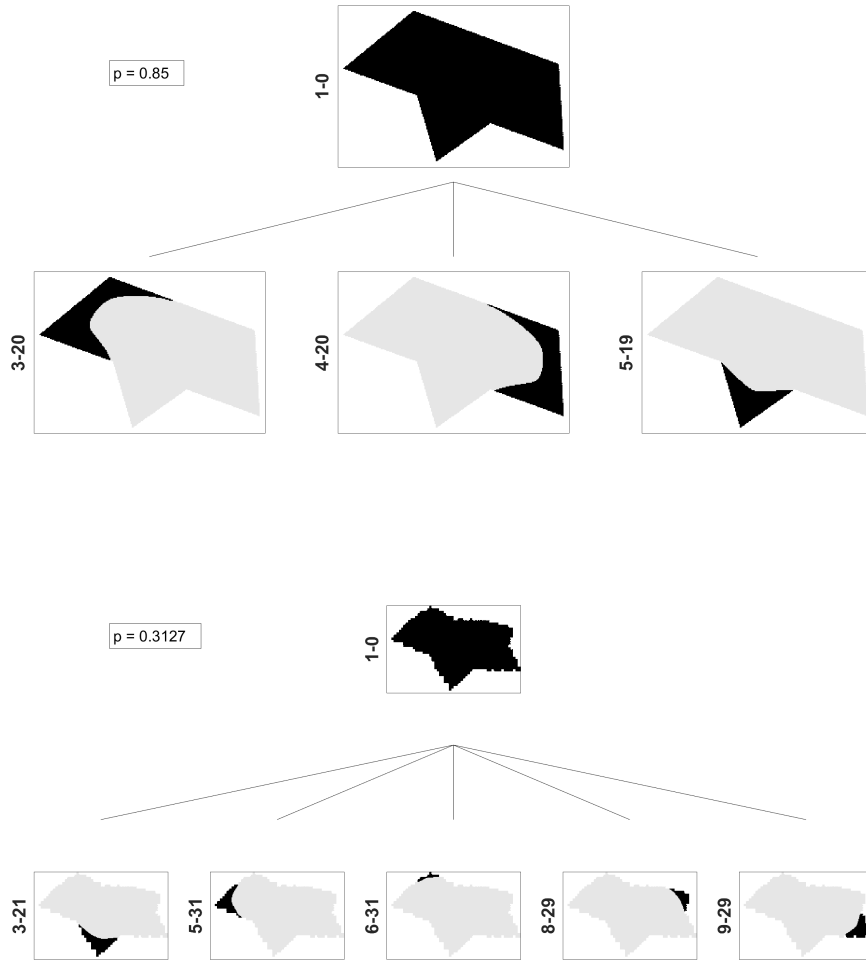


Figure 5.4: The individual representative hierarchy trees of the paper boats shown in Figure 5.3, i.e. the most probable tree samples generated by the stochastic reorganization process. The comparison shows how boundary deformations can affect the resulting individual representatives.



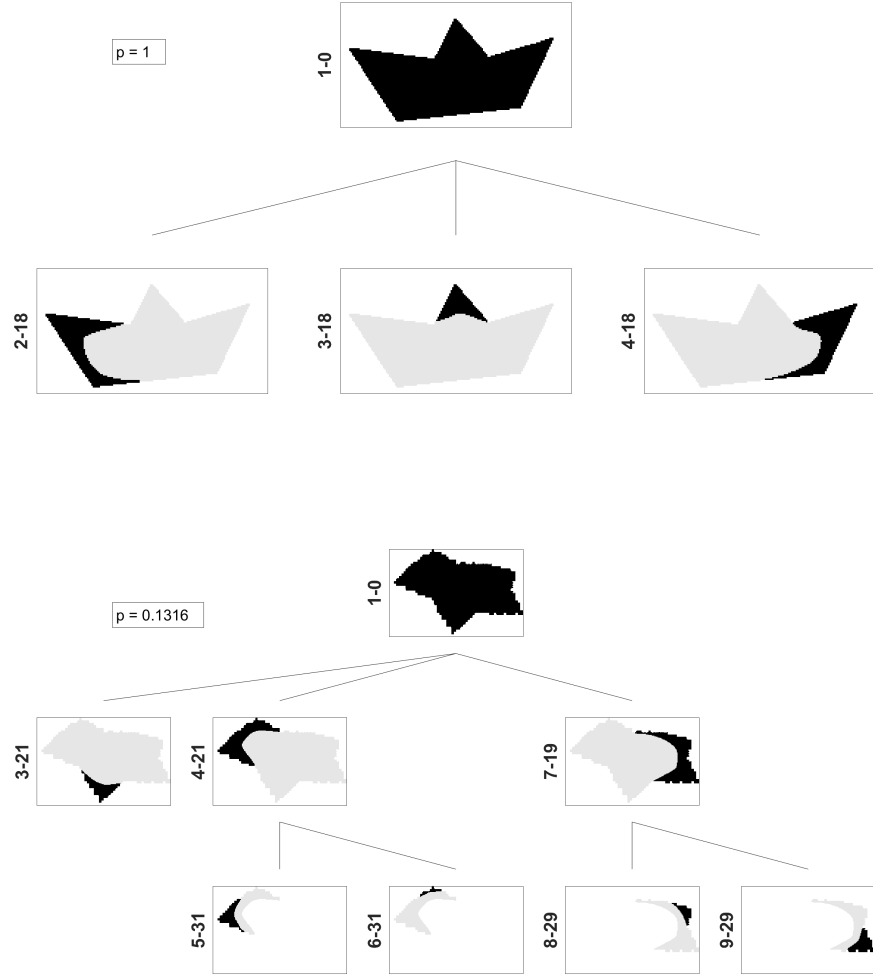


Figure 5.5: The collective decomposition hierarchy tree of the paper boat category (top), and the improved individual decomposition hierarchy of the wrinkled paper boat (bottom), which is the nearest tree to the collective hierarchy tree among all samples generated by the stochastic reordering process for the wrinkled paper boat. Observe also that the first level of the decomposition tree of the wrinkled boat becomes similar to that of the non-wrinkled boat, which was shown in Figure 5.4 to the top, as the further splits of the boat sides are deferred to the next level of the tree.

## 5.2 Experimental Results

We demonstrate shape decomposition performance our groupwise analysis method on various object categories from the 1000 shapes dataset [3]. The dataset involves 1000 object silhouettes from 50 categories (20 silhouettes per category). The resulting decomposition hierarchies of individual objects are shown in Appendix B. In the following paragraphs, we give examples that show how the proposed groupwise analysis method improves decomposition hierarchies of individual objects in presence of different conditions.

Decomposition hierarchy of an object can be affected by possible deformations on objects boundary. In Figure 5.3, we compare the decomposition of a paper boat to that of its wrinkled version. Observe that the decomposition hierarchy of the wrinkled version has an additional level of decomposition, where each side splits into its top and bottom part. This additional level of decomposition, then, results into an individual representative, where the boats side as a whole is absent in contrast to non-wrinkled version (see Figure 5.4). In Figure 5.5, we show the collective hierarchy tree of the paper boat category, and the improved individual hierarchy tree of the wrinkled version obtained with the help of the proposed groupwise analysis method. Observe that, consistent with the collective hierarchy, the sides of the boat are preserved in the improved hierarchy.

In Figures 5.6 and 5.7, we depicted the collective hierarchy tree of the cat category, the preliminary decomposition hierarchy of a specific cat object from that category, its individual representative tree and, finally, the optimized individual hierarchy tree obtained for that object, respectively. Observe in the individual representative that the head as a whole is missing and replaced by the ears and the nose due to the relative proximity between the split values of the parent (24) and its descendant nodes (34, 39). Then, the front legs are represented one level down the hind legs.

The partial occlusion of one front leg by the other leg results in relatively distant split values (24 vs. 45) which, most of the times, prevent replacement of the parent node (front legs unified) by its descendants (separated legs) in the reorganization process. In the optimized representation, though, both issues are resolved thanks to our collec-

tive analysis method: all legs are on the same level, and the head is preserved with its subparts. Indeed, if the ears and the nose were removed from the representation, the optimized hierarchy would have the same structure as the collective hierarchy with an exact mapping of counterparts. The issue of difference in the levels of the legs is also observed in the individual representative of the elephant depicted in Figure 5.8.

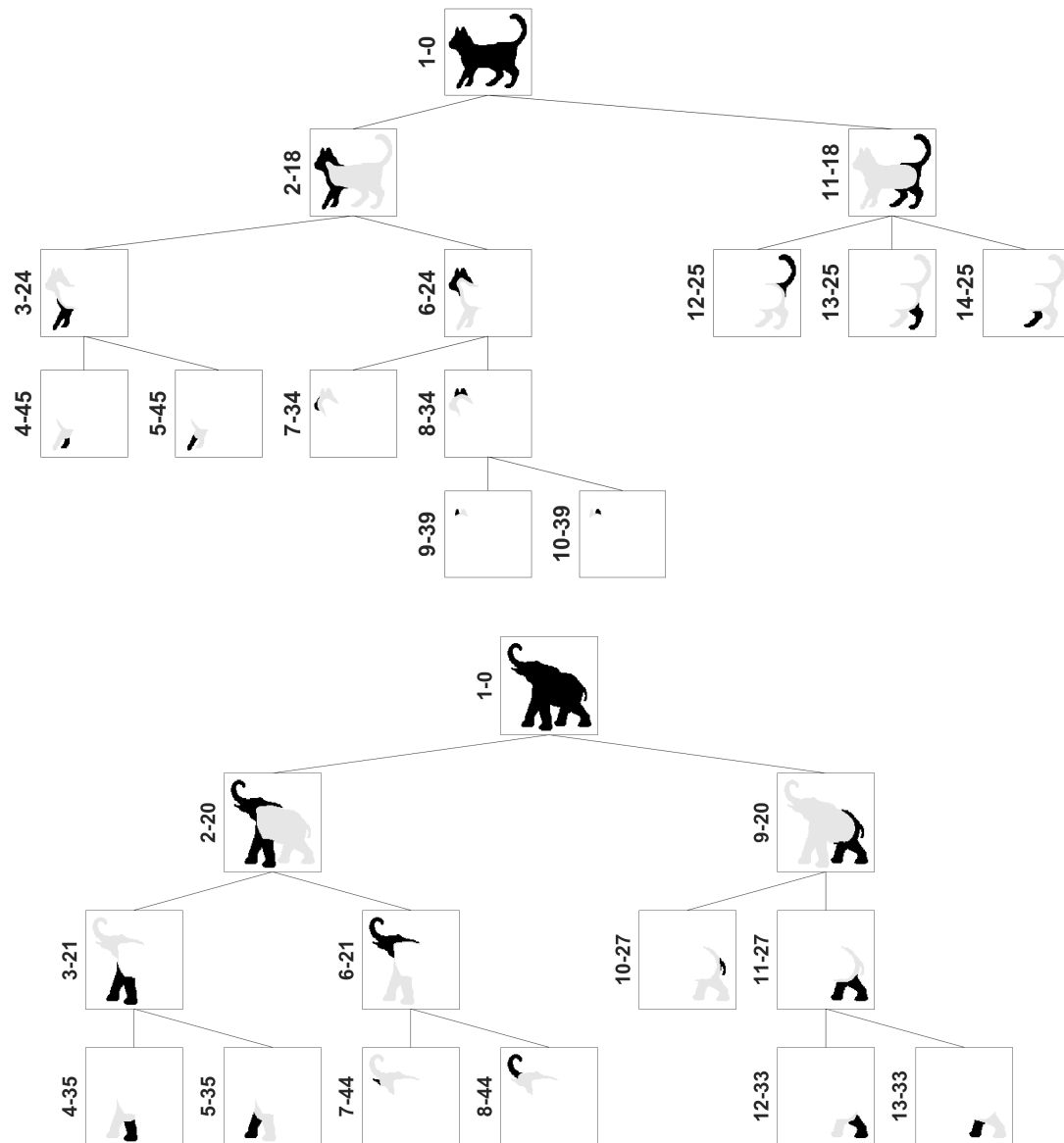


Figure 5.6: The preliminary decomposition hierarchy tree of a cat and elephant silhouette.

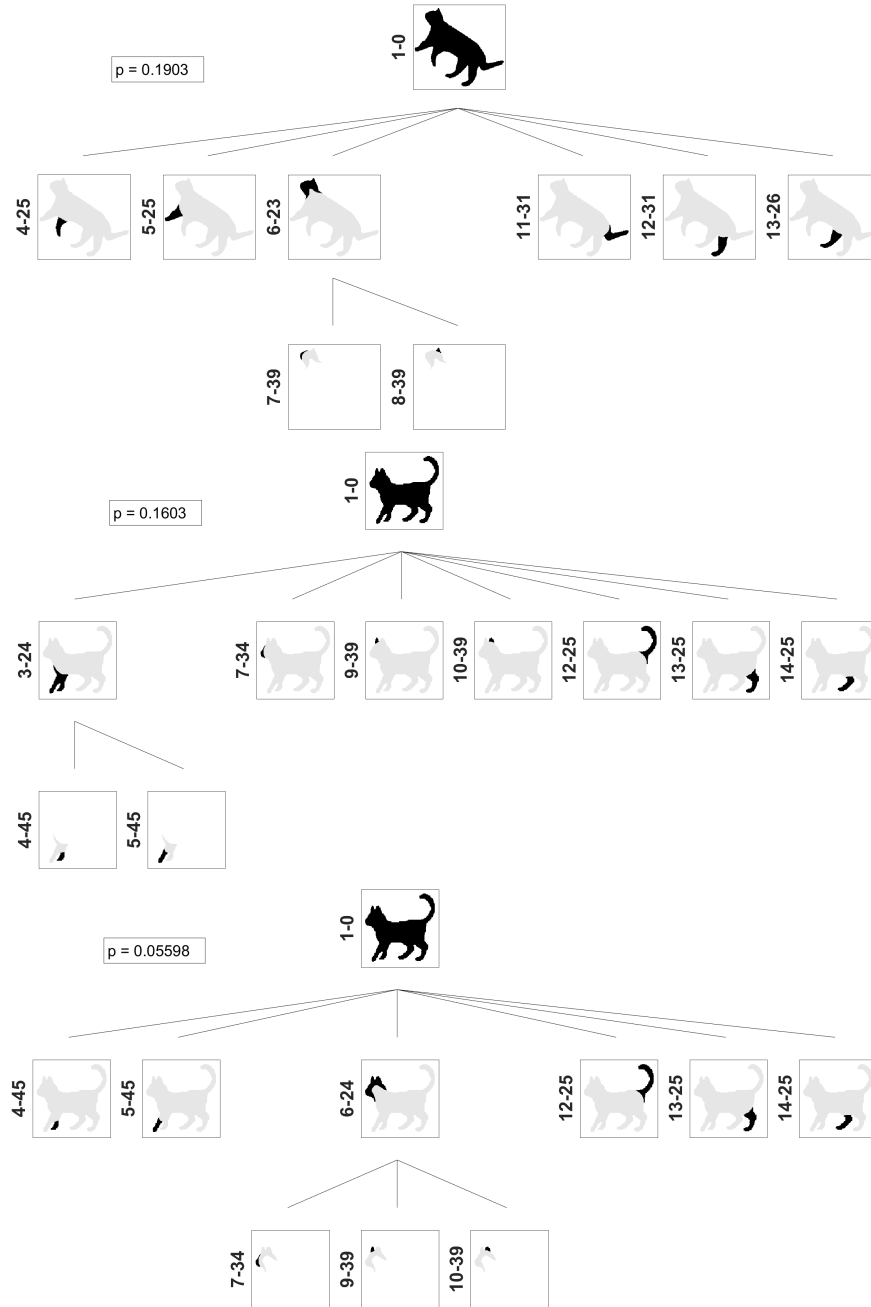


Figure 5.7: The collective decomposition hierarchy tree of the cat category (top), the individual representative tree (middle) of the specific cat whose preliminary tree is shown in Figure 5.6, and the optimized individual hierarchy (bottom) obtained by selecting the nearest tree sample to the collective tree.

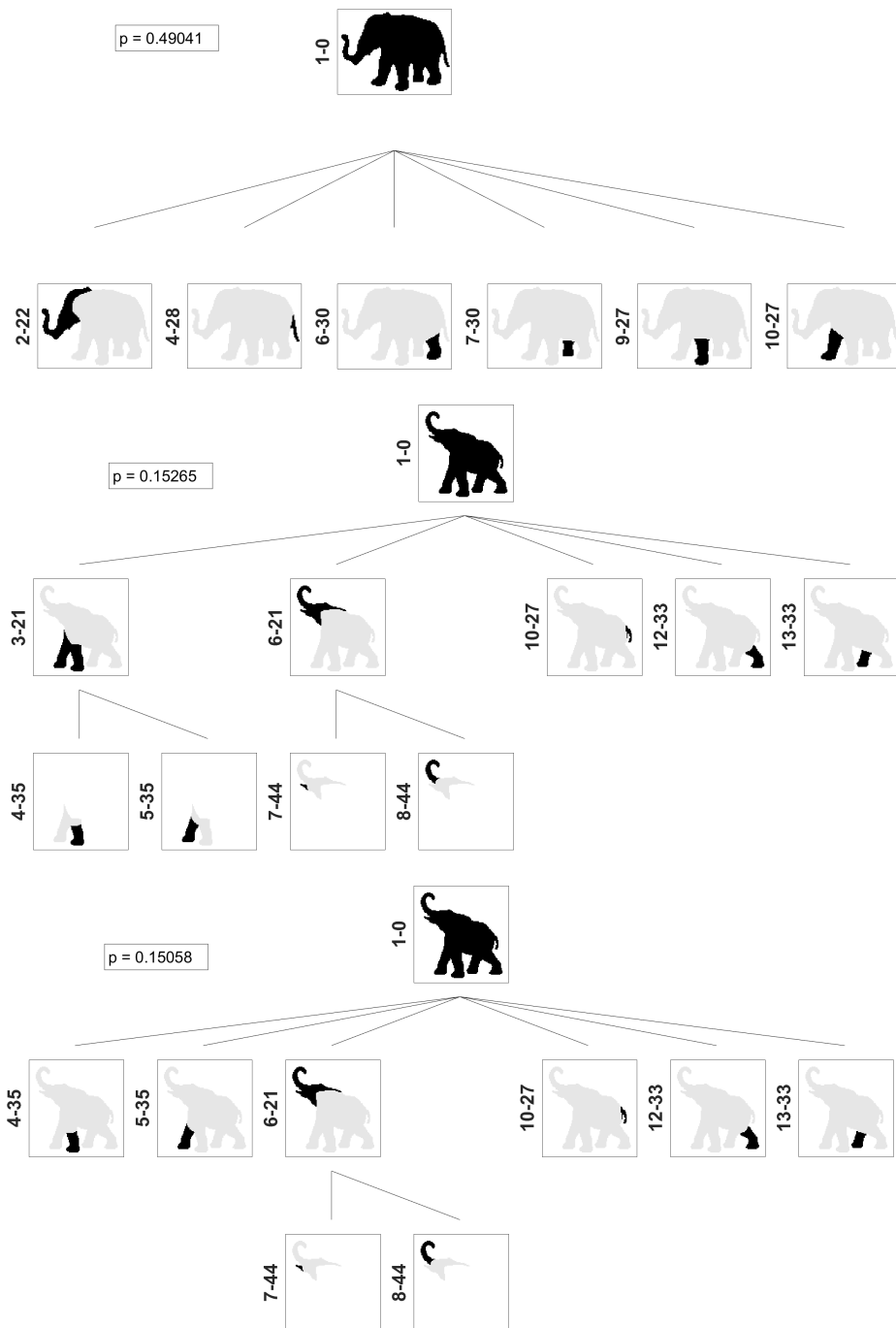


Figure 5.8: The collective decomposition hierarchy tree of the elephant category (top), the individual representative tree of a specific elephant object (middle), and the optimized individual hierarchy (bottom) obtained for that object.

## CHAPTER 6

### UNSUPERVISED SEGMENTATION OF 3D MESH MODELS

Real world objects are commonly modeled using polygon meshes which define the shape of the object by a collection of vertices, edges and faces. As in the case of additive manufacturing, we may need to segment a 3D mesh model into connected disjoint parts guided by the application specific objectives. Mesh segmentation also facilitates the following graphics applications. In metamorphosis [61, 80, 25], mesh segmentation is used for establishing correspondence. Then, it is utilized for improving compression rate in compression and simplification applications [37, 17]. In 3D modeling, segmented parts from existing object models are combined to synthesize new object designs [15]. Other applications include skeleton-driven animation [39], texture mapping [43], collision detection [44], and retrieval [81].

In this chapter, we developed an unsupervised algorithm for mesh segmentation that exploits part coding feature of SAP (projected onto the mesh surface) together with two local surface geometry cues, namely dihedral angle and concavity information.

In Section 6.2, we give the details of the segmentation algorithm. The results, given in Section 6.3, show that SAP has the greatest contribution to the algorithm performance. Moreover, in comparison with the state-of-the-art, the algorithm performs best among unsupervised algorithms for single mesh segmentation (even better than a human) and it also performs comparable to a supervised algorithm and a segmentation method for joint segmentation of multiple meshes.

## 6.1 Related Work

Started with the earliest works, local (per-vertex/facet or pairwise) geometric properties of the surface have been exploited in segmenting meshes. Clustering based methods [61, 39] define pairwise distances between surface points using geodesic and angular distances and concavity information. Region growing based methods [52, 55] partition the mesh into surface patches with consistent curvatures bounded by contours of curvature extrema.

Surface properties such as principal curvatures, angular distance, and concavity provide low-level information to locate segment boundaries as well as to measure the geometrical consistency of the segment, yet bring sensitivity to local surface deformations induced by noise, small creases and pose changes. Geodesic distance provides spatial coherence and is invariant under nearly-isometric deformations such as pose change; however, it often causes spurious divisions of geometrically coherent parts due to polarization effect. Furthermore, it is sensitive to changes in topology. Gaussian curvature (product of principal curvatures) is another isometry invariant, yet its approximation on a discrete mesh surface is prone to noise.

Higher-level information can be derived by considering larger neighborhoods to measure geometric properties in multiple scales in order to alleviate sensitivity to local perturbations on the shape surface. To this end, principal component analysis is employed both over surface- [35] and volume(ball)-neighborhoods [76, 41]. Furthermore, in [35, 51], curvature is measured on patches of growing size fitted to local surface points. Computed using spectral properties of Laplace-Beltrami, Heat kernel signature (HKS) [65] also provides multi-scale curvature related information at each surface point.

Average Geodesic Distance (AGD) by [28] represents an extreme case that takes global interactions over the surface into account. AGD characterizes a surface point within the context of global mesh geometry by aggregating pairwise geodesic distances from the point to all other surface points. Topology driven methods [38, 6], and saliency-guided segmentation approach [46] make use of AGD to detect feature points through which the intrinsic shape structure is captured to guide the segmenta-



tion process. Conformal factors (CF) [5] is another global representation that is stored locally. It is derived from local Gaussian curvature, hence invariant to pose changes. Both AGD and CF are sensitive to topological differences.

Intrinsic symmetry information is captured using approximate Killing vector fields [4] to discover shape primitives that are the basic parts of the shape which might have undergone nearly-isometric deformations due to pose changes but preserving intrinsic symmetry [63].

Spectral methods [48, 78, 73, 71] define the Laplacian on the mesh (or dual) graph whose weights encode pairwise similarities between adjacent nodes based on various properties of the surface geometry. Eigenvectors and eigenvalues of the graph Laplacian matrix reveal global shape properties. [73] employs a further analysis of individual eigenvectors from which a concavity-aware feature called single segmentation field (SSF) is synthesized. In [71], the spectral analysis is driven by an initial over-segmentation via construction of a heterogeneous graph that combines local face affinities with nonlocal patch affinities.

Line-of-sight visibility is utilized to capture long range interactions among surface points through the shape volume. A point is considered in the line-of-sight of another point if they are connected through a line which lies entirely within the shape volume. [60] propose an invariant under pose change, namely Shape Diameter Function (SDF), that approximates the local thickness of the shape volume. SDF is measured by propagating rays from a surface point around the inward normal direction, and then averaging distances to sighted points on the opposite side of the surface. Despite the altered visibility conditions due to pose changes, the proposed function remains largely invariant by setting the computational parameters carefully and smoothing the measurements as a post-processing. Based on the observation that the points on the surface of a convex part are visible to each other, [2, 34] cluster the surface points that are mutually visible and geometrically connected in order to segment the mesh into weakly convex parts. [47] proposes Continuous Visibility Feature (CVF) which characterizes a surface point with respect to its sight both through the shape volume and over the shape surface. A serious drawback of CVF is its sensitivity to concavities on the surface due to its stronger visibility definition.

Given a task that involves segmenting a set of similar objects without prior knowledge, a practical way of thinking is to enable knowledge transfer by means of joint segmentation to achieve set-wide consistency. Knowledge transfer allows identification of salient segments shared across the objects and provides external support to assist segmentation of challenging objects on which derived shape properties are inadequate to recover salient segments. However, accurate identification of part-correspondences between objects in the presence of large intra-class variations in geometry and scale of semantically similar parts pose a challenge to joint segmentation processes. To factor out the scale variation, [75] proposes to pre-cluster the objects into groups with homogeneous part scaling and employs the global rigid alignment based algorithm [23] in each group independently. Descriptor space clustering based joint segmentation algorithms [62, 32, 31, 50], in essence, first over-segment individual meshes into patches, then measure patch proximities based on several shape descriptors between all patches from all meshes, and finally cluster the patches to have consistent segmentations and part correspondences simultaneously. The patch descriptors used for proximity measurements include patch area, patch-wide geometric surface properties (including histograms of per-vertex/facet properties, principal component analysis based features and AGD), visibility-based SDF, HKS, CF, and geodesic distance to the shape base.

A nice instrument for consistency is a voting mechanism that pools multiple segmentation results, yielding a continuous function defined over the edges of a mesh that indicates the probability that an edge is cut by a segmentation [22]. The function is used to evaluate a segmentation by weighting the cut to obtain a consistency score.

## 6.2 Method

The proposed algorithm consists of three steps. We provide its pseudo-code in Algorithm 2 and explain the details of each step in the following paragraphs.

**Spectral analysis** Spectral analysis of the dual Laplacian matrix has been successfully applied to capture nonlocal shape features using local features of the surface geometry such as concavity, curvature, geodesic distance etc. Our contribution in this

---

**Algorithm 2:** Segmentation for a given number of segments

---

**input** : Mesh  $\mathcal{M}$ , representation  $SAP$ ,

number of segments  $k$

**output:** Segment labels  $l$

**begin**

$\{v_i : i = 1..k - 1\} \leftarrow \text{Spectral analysis}(\mathcal{M}, SAP);$

$\{p_\tau : \tau \in \mathcal{M}\} \leftarrow \text{Soft-clustering}(\{v_i\}, k);$

$\{l_\tau : \tau \in \mathcal{M}\} \leftarrow \text{Hard-segmentation}(\{p_\tau\});$

**end**

---

part is integration of SAP into the definition of the pairwise distance measure between neighboring mesh elements to be used in constructing the Laplacian.

Let  $G = (V, E, w)$  be the dual graph of the mesh  $\mathcal{M}$  with the set of nodes  $V = \{\tau : \tau \text{ represents a triangle of } \mathcal{M}\}$ , the set of edges  $E = \{(i, j) : \tau_i \text{ and } \tau_j \text{ are adjacent}\}$  and the function of edge weights  $w : E \rightarrow R^+$ . The edge weight  $w_{(i,j)}$  associated with the edge  $(i, j)$  represents the similarity between the adjacent pair of triangles  $\tau_i$  and  $\tau_j$  and computed as shown in (6.1).

$$w_{(i,j)} = \frac{|e_{ij}|}{|e|} \exp \left\{ -\frac{\delta(\tau_i, \tau_j)}{\bar{\delta}} \right\} \quad (6.1)$$

where  $|e_{ij}|$  is the length of the shared edge normalized against the mean length  $|e|$  of all edges in  $\mathcal{M}$ ,  $\delta(\tau_i, \tau_j)$  is the distance between the pair, measured using (6.2) and normalized by the average distance  $\bar{\delta}$  over all pairs of adjacent triangles.

$$\delta(\tau_i, \tau_j) = \kappa(i, j) \times (\delta_{SAP}(i, j) + \delta_{DHD}(i, j)) \quad (6.2)$$

$\delta_{SAP}(i, j)$  denotes the difference between the SAP values assigned to the distinct vertices of  $\tau_i$  and  $\tau_j$ .  $\delta_{DHD}(i, j)$  is an angular difference based on the dihedral angle  $\phi_{i,j}$ —the angle between outward normals—and formulated by  $\min(3 \times \phi_{i,j}/\pi, 1.0)$  to emphasize low-frequency geometric features.  $\kappa(i, j)$  is a constant to apply the minima rule which suggests higher priority for concave creases. We set  $\kappa(i, j) = 1.0$  when the edge between  $\tau_i$  and  $\tau_j$  is concave, and 0.1 otherwise.

We construct the dual Laplacian matrix  $L$  as follows:

$$L_{ij} = \begin{cases} -w_{(i,j)} & (i,j) \in E \\ \sum_k w_{(i,k)} & i = j, (i,k) \in E \\ 0 & \text{otherwise} \end{cases} \quad (6.3)$$

We compute a sequence of eigenvectors  $v_1, \dots, v_n$  of the Laplacian matrix corresponding to the first  $n$  nonzero eigenvalues of smallest magnitude  $\lambda_1, \dots, \lambda_n$  in ascending order.

Salient parts are strongly separated in the feature space associated with the eigenvectors, yet some of the eigenvectors may represent small surface features isolated from their surrounding region because of strongly discriminating cues. To cope with such cases, we employ a pruning step in which the mesh is bisected on the basis of each individual eigenvector (by thresholding) and select first  $k - 1$  eigenvectors for which the size of the smaller part is greater than a given size threshold *relSize* which we set to 1% of the entire surface area.

**Soft-clustering** We soft-cluster the mesh triangles into  $k$  clusters, i.e. we compute cluster posterior probabilities for each triangle, based on their distribution in the feature space. We represent clusters as components of a Gaussian Mixture Model (GMM), whose parameters are estimated by employing expectation-maximization algorithm. Given a triangle  $\tau$ , the associated feature vector  $f_\tau$  and the estimated model parameters  $\theta_i$  for  $(i = 1, 2, \dots, k)$ , we compute posterior probabilities vector  $p_\tau = [P(\theta_1|f_\tau), \dots, P(\theta_k|f_\tau)]^T$ .

**Hard-segmentation** Assigning a mesh triangle to the most probable cluster with respect to the posterior probabilities appears to be a reasonable solution for many shapes. For many other shapes, however, it can fail in producing smooth and plausible segment boundaries that adhere to concave creases of the mesh surface. To this end, following [60], we define an optimization problem that search for a final labeling that simultaneously tries to assign each triangle to a cluster with higher posterior probability, and to yield smooth and plausible segment boundaries. Specifically, we

consider the following energy minimization problem:

$$\begin{aligned} \arg \min_l \quad & \sum_{\tau \in \mathcal{M}} F(\tau, l_\tau) + \beta \sum_{(i,j) \in E} Q(l_{\tau_i}, l_{\tau_j}) \\ F(\tau, l_\tau) = & -\log(P(\theta_{l_\tau} | f_\tau) + \epsilon) \\ Q(l_{\tau_i}, l_{\tau_j}) = & \begin{cases} -\log(\kappa(i, j) \times \phi_{i,j}/\pi) & l_{\tau_i} \neq l_{\tau_j} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (6.4)$$

where  $F$  is the fidelity term that measures the disagreement between the labeling  $l$  and the GMM that was fit to data, and  $Q$  is the quality term that imposes short/smooth boundaries passing through concave creases. The parameter  $\beta$  determines the weight of the quality against fidelity.

We employ the expansion move algorithm based on graph cuts proposed in [8] to solve the energy minimization problem.

### 6.3 Results and Benchmark Evaluation

We evaluate the performance of our algorithm with respect to state-of-the-art methods by computing the Rand index performance metric on the Princeton Segmentation Benchmark set [11]. The benchmark dataset contains 4300 human generated segmentations for 380 watertight mesh models. The mesh models are adapted from the Watertight Track of the 2007 SHREC Shape-based Retrieval Contest [21] and uniformly distributed to 19 different object categories that range from human and animal bodies to mechanical CAD parts and furnitures. Two of them (armadillo and pliers) contain various appearances (different poses with missing parts occasionally) of the same object. Another seven categories (human, ant, octopus, teddy bear, hand, bird, four leg) consist of different articulated objects of the same type that appear in different poses. There are also categories (cup, vase, chair) where the objects have non-zero topological genus.

During the evaluation of segmentation algorithms, in order to provide a fair comparison between the algorithms that require the number of segments from the user and those that automatically find it, the benchmark chooses a separate value for the number of segments for each model by setting it to the mode of the number of segments

that appear in human generated segmentations of that model. This models a typical use case scenario where the human user looks at the model and then runs the segmentation algorithm with an estimation of the number of segments. Note that giving the exact number of segments for each human generated segmentation would unfairly benefit those algorithms that require the number of segments.

State-of-the-art methods have reported performance results using the Princeton Benchmark Rand Index measure making it the defacto standard for comparison of the mesh segmentation algorithms. The region-based Rand index was originally proposed for measuring similarity between two data clusterings in [59]. In our context, it measures the likelihood that a pair of mesh faces are either in the same segment or in different segments in both segmentations. The distinguishing property of the metric is that it can model overlapping regions without finding associations between segments.

Let  $S_1$  and  $S_2$  be two segmentations,  $s_i^1$  and  $s_i^2$  be the segment ids of the face  $i$  in  $S_1$  and  $S_2$ , and  $N$  be the total number of mesh faces. Then the Rand index is calculated using the formula:

$$RI(S_1, S_2) = \frac{\sum_{i,j,i < j} [P_{ij}Q_{ij} + (1 - P_{ij})(1 - Q_{ij})]}{\binom{N}{2}} \quad (6.5)$$

where  $P_{ij} = 1$  if and only if  $s_i^1 = s_j^1$  and  $Q_{ij} = 1$  if and only if  $s_i^2 = s_j^2$ . The first term of the sum ( $P_{ij}Q_{ij}$ ) counts the pairs  $i$  and  $j$  that fall to the same segment in both segmentations, whereas the second counts those that are in different segments in both segmentations. The denominator of the formula is simply the total number of pairs. We note that the Rand index metric reported by the Princeton benchmark is in fact calculated as  $1 - RI(S_1, S_2)$ , i.e. it measures the dissimilarity between two segmentations rather than the similarity which is measured by  $RI(S_1, S_2)$ . Therefore, in our report, lower Rand index values indicate better performance.

The benchmark evaluates the final Rand index performance of an algorithm as follows. First, Rand index values for each human generated segmentation of a model is computed by comparing it to the resulting segmentation of the algorithm (see (6.5)). Then, those values are averaged to obtain the Rand index value of the model. Finally, Rand index values of all models and models from the same category are averaged to determine the overall and the category-based performance of the algorithm.

In Figure 6.1 (overall) and in Table 6.1 (per class), Rand index performances of several algorithms from different categories are compared with human performance (small values indicate better performance). Unsupervised single mesh segmentation category includes our algorithm, spectral mesh segmentation using SSF (SSF, [73]), Mumford–Shah mesh decomposition (M-S, [78]) and another spectral algorithm driven by heterogeneous graphs (HGD, [71]). Other methods include Joint Shape Segmentation (Joint, [32]) for segmenting multiple meshes jointly and the supervised method for segmentation and labeling [35] with varying number of samples used for training: 95% of the samples for each category (Sup95), 60% (Sup60), and 15% (Sup15). Our method performs the best among state-of-the-art algorithms from the same category, performs comparable to the joint segmentation method that exploits information from entire dataset, surpasses the supervised method when the size of the training set is specified as 60% of each category (Sup60), and appears to be even better than a human subject. Representative segmentations from benchmark evaluation of our algorithm are shown in Appendix A.

We also evaluated the contribution of SAP to the benchmark performance of the proposed algorithm by redefining the distance measure (6.2) so that it involves different combinations of the cues. The resulting benchmark performances using the modified distance measures are shown in Figure 6.2. Obviously, the best individual performance is achieved using SAP, which already competes with unsupervised single mesh segmentation algorithms. Moreover, the algorithm outperforms them using SAP and concavity without dihedral angle.

In Figure 6.3, illustrative segmentations obtained using the distance measure (6.2) both with and without SAP are depicted. The contribution of SAP is observed in different ways. SAP acts as a supportive cue when the surface-based cues remain weak to segment a protrusion correctly, e.g., the angular distance is small on a remarkable portion of the segment boundary. Thereby our algorithm manages to separate the pair of glasses and to segment the left leg of the walking woman (instead of a finger), the dorsal fin of the fish, the nose of the bear and the bust, the left wing of the airplane, the middle finger of the hand, and the handle of the vase. SAP also improves consistency in segmenting different poses of shapes. In the walking and running women example, without using SAP, the algorithm segments the right hand instead of the right leg;

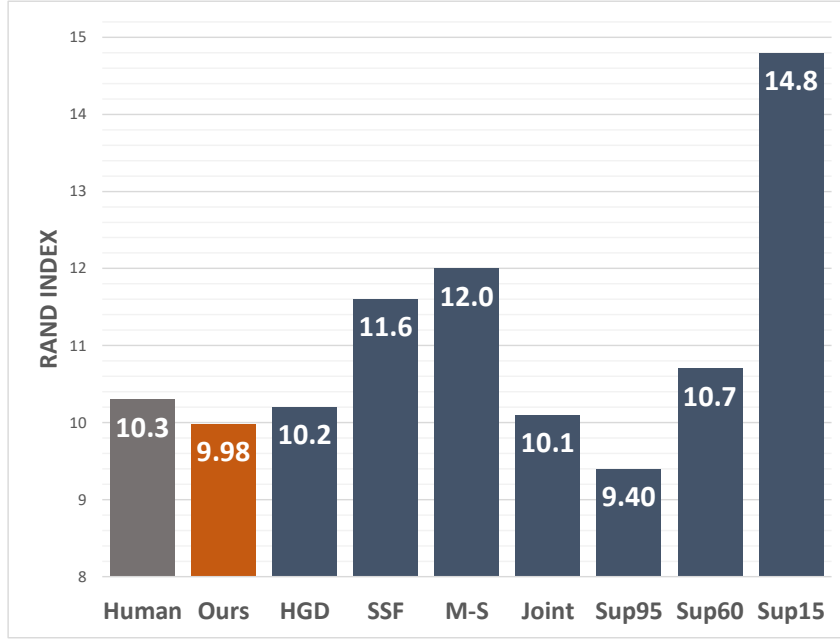


Figure 6.1: Benchmark performance of our algorithm (Ours) in comparison with state-of-the-art methods of different categories: unsupervised single mesh segmentation algorithms (HGD, [71]), (SSF, [73]) and (M-S, [78]), algorithm for joint segmentation of multiple shapes (Joint, [32]), supervised segmentation with various portions of samples used for training (95%-Sup95, 60%-Sup60, 15%-Sup15, [35]).



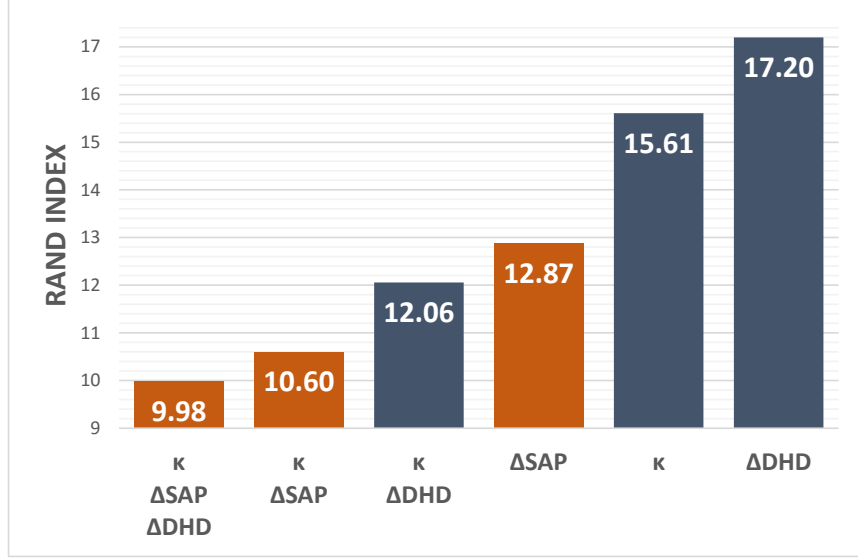


Figure 6.2: Evaluation of SAP’s contribution to the benchmark performance of our algorithm based on Rand Index values. The contributions are evaluated by using various combinations of the cues in the definition of the distance measure (6.2) as indicated by the labels, i.e. the leftmost bar represents the performance using all cues: SAP function ( $\delta_{SAP}$ ), dihedral angle ( $\delta_{DHD}$ ) and concavity information ( $\kappa$ ).

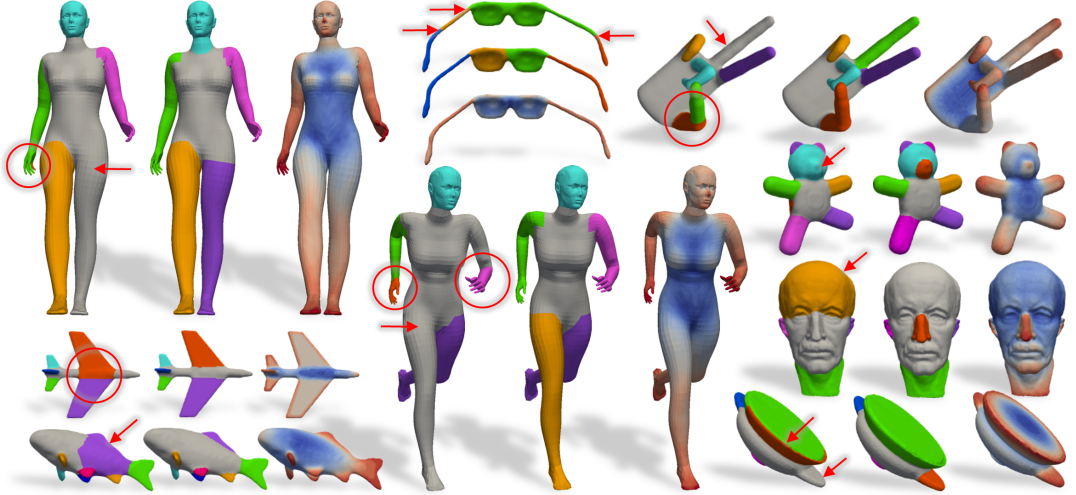


Figure 6.3: SAP contributes to segmentation performance in different ways. In each triplet, from left to right, we show segmentation results using and not using SAP, and SAP itself.

Table 6.1: Princeton Benchmark Rand Index Values

		Single-Unsupervised							Joint	Supervised		
	Human	Ours	HGD	SSF	M-S	CVF	RC	SDF	JSS	SB19	SB12	SB3
Average	10.3	<b>9.98</b>	10.2	11.6	12.0	15	15.3	17.6	10.1	9.4	10.7	14.8
Human	13.5	13.67	12.4	12.8	<b>11.1</b>	14	13.1	17.9	11.3	11.9	12.9	14.7
Cup	13.6	11.61	<b>11.0</b>	14.6	20.4	23	21.9	35.8	11.2	9.9	9.9	10.0
Glasses	10.1	<b>9.22</b>	9.9	11.3	9.4	19	10.1	20.4	9.9	13.6	14.1	14.2
Airplane	9.2	<b>8.50</b>	11.2	13.2	11.1	20	12.2	9.2	10.2	7.9	8.2	10.2
Ant	3.0	<b>1.80</b>	2.0	2.8	2.2	4	2.5	2.2	2.4	1.9	2.2	2.6
Chair	8.9	<b>6.33</b>	7.4	8.4	10.9	7	18.4	11.1	9.6	5.4	5.6	6.6
Octopus	2.4	<b>1.78</b>	2.6	2.6	2.5	3	6.3	4.5	7.2	1.8	1.8	2.2
Table	9.3	6.24	7.0	<b>6.1</b>	10.3	9	38.3	18.4	6.6	6.2	6.6	11.1
Teddy	4.9	3.36	3.9	3.6	<b>3.2</b>	7	4.5	5.7	3.3	3.1	3.2	5.6
Hand	9.1	9.65	10.7	11.0	<b>7.9</b>	13	9.0	20.2	13.1	10.4	11.2	15.8
Plier	7.1	8.05	<b>5.7</b>	8.5	8.9	22	10.9	37.5	7.5	5.4	9.0	10.5
Fish	15.5	19.33	18.6	21.5	29.6	<b>18</b>	29.7	24.8	13.1	12.9	13.2	13.5
Bird	6.2	<b>6.58</b>	7.8	7.8	9.4	18	10.6	11.5	7.6	10.4	14.8	18.6
Armadillo	8.3	<b>8.63</b>	10.3	9.1	8.7	10	9.2	9.0	7.4	9.0	8.4	8.6
Bust	22.0	25.29	25.8	28.6	25.1	30	<b>23.2</b>	29.8	19.8	21.4	22.2	39.3
Mech	13.1	11.13	<b>10.5</b>	12.6	13.1	14	27.7	23.8	13.3	10.0	11.8	24.0
Bearing	10.4	<b>9.29</b>	9.5	14.8	16.6	13	12.4	11.9	11.3	9.7	17.6	32.7
Vase	14.4	12.71	<b>12.1</b>	15.4	12.5	18	13.3	23.9	13.2	16.0	17.1	25.3
Fourleg	14.9	16.57	15.7	16.5	<b>14.4</b>	17	17.2	16.1	11.2	13.3	13.9	16.3

moreover, the upper half of the left arm is merged with the body. It is shown that with the help of SAP, the algorithm segments the walking and running women consistently.

**Limitations** Despite the benchmark performance of our algorithm, there are two issues that have to be dealt with. One of them is missing significant parts due to current fixations of the parameters. In Figure 6.4, two such cases are depicted. In the first case, right engine of the airplane is removed in the hard-segmentation step due to the large weight ( $\beta$ ) given to the boundary quality term in (6.4). The part is recovered by choosing a smaller weight. In the second case, the beak of the bird is missed because of its size, which is below the relative size threshold and hence the associated eigenvector is ignored in the spectral analysis step. Similarly, a smaller selection of the parameter recovers the beak. Missing part issue suggests user adjustable parameters or employing a means of adaptive parameter selection.

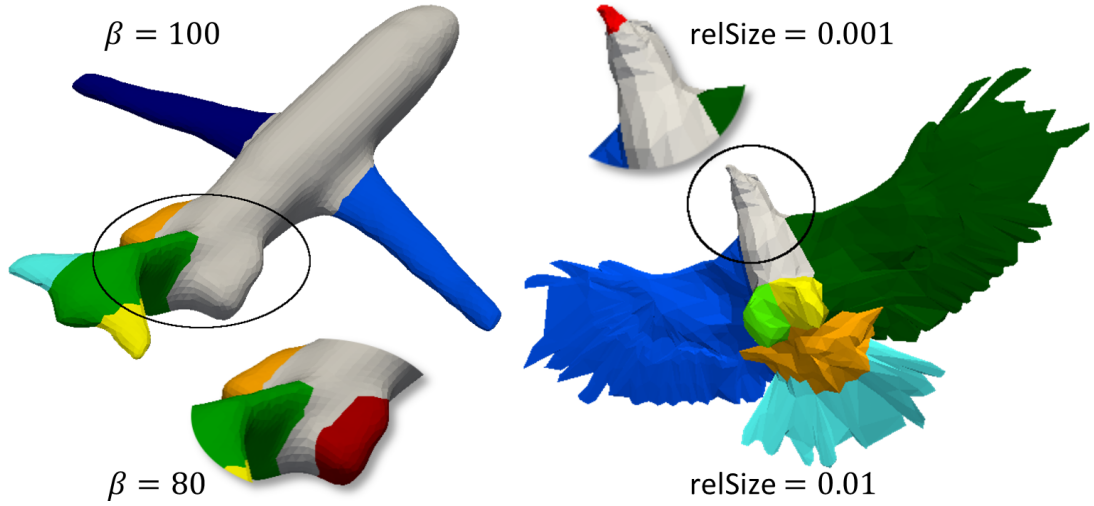


Figure 6.4: Missing parts due to the current parameter fixations. By adjusting the parameters, we can recover the missing parts.

Our algorithm also shares a common issue with spectral analysis based methods which is determining the eigenvectors to be used for segmentation. Our algorithm selects  $k - 1$  eigenvectors to segment the mesh into  $k$  parts following the approach in [78] which, however, causes instabilities especially in decomposing complex shapes such as the ones from human or four-legs categories into finer segments. The problem could be handled by an adaptive selection mechanism similar to the one proposed in [73] based on second-order difference analysis of the Laplacian matrix eigenvalues.



## CHAPTER 7

### SUMMARY AND CONCLUSION

In this thesis study, we developed methods for shape analysis in two and three dimensions. Toward this end, we first presented an efficient computational model for shape analysis. The model solves Poisson's equation in the shape domain with specially designed right-hand sides to explore the entire shape yielding a family of part-coding functions.

We showed that the proposed model is a superior alternative to a previous nonlocal model from the aspects of computational efficiency, part-coding capability and applicability. Our model is computationally efficient, since it reduces to solving a linear system of equations with a sparse and positive definite matrix for which efficient solvers exist. In contrast, the previous model yields a full system matrix for which a direct solution becomes unfeasible as the shape domain gets larger, and hence, iterative solvers are employed to obtain approximate solutions. Then, we showed that by varying the parameter of our model, we obtain multiple decompositions of shape, extending the part-coding capability of the previous model. Specifically, our model relaxed the division of the shape domain into central (positive valued) and peripheral (negative valued) regions which was strict in the previous model. We demonstrated that such a relaxation enables adjusting the granularity of parts encoded in the solution within the peripheral region. And finally, we demonstrated that, thanks to the local formulation, our model is also applicable to image frames in which shapes are only implied by incomplete edge fragments, whereas the nonlocal previous model requires the complete knowledge of the shape domain.

We further aggregated multiple part-coding functions sampled from our model into an integer-valued function defined on the shape, namely Source-Aggregated-Poisson

(SAP). We showed that SAP encodes evolution of the shape boundary (represented by the boundary of the positive set throughout those functions) induced by morphological and curvature dependent factors. Then, we exhibited robustness of SAP under various visual transformations, occlusions and topological distortions.

In order to demonstrate the claimed part-coding capability of our model, and hence of SAP, we first devised a simple shape decomposition method that hierarchically partitions the shape domain using level sets of SAP. Then, we proposed a probabilistic tree structure to effectively and flexibly represent multiple interpretations of decomposition hierarchy. The proposed structure was an ordinary tree (i.e. the one produced by the simple decomposition method) endowed with a probabilistic process that traverse the tree and change the local tree configuration with a probability imposed by the SAP levels attributed to the related tree nodes (parent and its children). Finally, in the context of a groupwise analysis task, we demonstrated how such a probabilistic structure helps optimizing representations of individuals in the group of similar shapes in an attempt to reduce within group representational variance. For this purpose, we first determined the collective representation of the shape group as the medoid of the most probable hierarchy tree samples that can be generated by the individual probabilistic hierarchy trees. Then, for each individual shape, we select the tree sample that is most similar to that collective representation.

Finally, in the context of mesh segmentation problem, we showed that SAP values near shape boundary proves to be useful in meaningful segmentation of the boundary, which is commonly represented by a surface mesh in 3D. To that end, we developed an unsupervised mesh segmentation algorithm that consists of three steps. In the first step, we combined volume based SAP values projected onto surface mesh with local surface geometry information to construct the dual graph Laplacian matrix and computed its spectrum. Then in the second step, we soft-clustered mesh elements in the feature space constructed using the computed spectrum, where clusters were represented by the components of a Gaussian Mixture Model and the output was the cluster association scores computed for each element. In the last step, we solved an optimization problem to assign each mesh element to a cluster with a high association score as possible while the segment boundaries are smooth and plausible.

We evaluated the performance of our mesh segmentation algorithm against state-of-the-art methods via the Princeton Segmentation Benchmark. Using the Rand Index error, the benchmark essentially measures the deviation of segmentations produced by our algorithm from the segmentations generated by an average human subject. Our algorithm performed the best among all unsupervised methods, and even comparable to joint and also supervised ones. Moreover, without the surface-geometry-based information and using only our volume-based SAP function projected to the surface, the algorithm could compete with other unsupervised algorithms. Another remarkable result was that among all segmentation cues SAP distinguishes as the one that contributes the most to the final performance of the algorithm. We showed that SAP improves the segmentation when surface geometry does not provide sufficient information for segmenting a part and should be supported by a volume-based cue. We also demonstrated that SAP helps improving consistency in segmenting different poses of objects with moving articulations or limbs. These results suggest that a meaningful segmentation of the object boundary demands more information than the local geometry of the boundary can provide. SAP as a volume-based nonlocal shape information has the potential to fill the gap.

From the part-coding functions of our model to derived SAP and decomposition hierarchies, all constructions of this thesis work are developed to represent the shape at different stages of visual processing. The natural continuation work will progress towards designing a vision system that would take the scene as its input and uncover the semantics via recognizing the objects and their relations.





## REFERENCES

- [1] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 408–416, New York, NY, USA, 2005. ACM.
- [2] S. Asafi, A. Goren, and D. Cohen-Or. Weak convex decomposition by lines-of-sight. *Computer Graphics Forum*, 32(5):23–31, 2013.
- [3] E. Baseski, A. Erdem, and S. Tari. Dissimilarity between two skeletal trees in a context. *Pattern Recognition*, 42(3):370 – 385, 2009.
- [4] M. Ben-Chen, A. Butscher, J. Solomon, and L. Guibas. On discrete killing vector fields and patterns on surfaces. *Computer Graphics Forum*, 29(5):1701–1711, 2010.
- [5] M. Ben-Chen and C. Gotsman. Characterizing shape using conformal factors. In *Proceedings of the 1st Eurographics Conference on 3D Object Retrieval*, 3DOR '08, pages 1–8, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.
- [6] S. Berretti, A. D. Bimbo, and P. Pala. 3d mesh decomposition using reeb graphs. *Image and Vision Computing*, 27(10):1540 – 1554, 2009. Special Section: Computer Vision Methods for Ambient Intelligence.
- [7] M. Botsch, D. Bommers, and L. Kobbelt. Efficient linear system solvers for mesh processing. In R. Martin, H. Bez, and M. Sabin, editors, *Mathematics of Surfaces XI*, volume 3604, pages 62–83. Springer, 2005.
- [8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.
- [9] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, volume 2, pages 60–65. IEEE, 2005.

- [10] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, 2008.
- [11] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Trans. Graph.*, 28(3):73:1–73:12, 2009.
- [12] E. H. Cohen and M. Singh. Geometric determinants of shape segmentation: Tests using segment identification. *Vision Research*, 47(22):2825–2840, 2007.
- [13] H. Dutagaci, C. P. Cheung, and A. Godil. Evaluation of 3D interest point detection techniques via human-generated ground truth. *The Visual Computer*, 28(9):901–917, 2012.
- [14] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Trans. Graph.*, 21(3):249–256, 2002.
- [15] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH ’04, pages 652–663, New York, NY, USA, 2004. ACM.
- [16] Y. Gao and S. Bouix. Statistical shape analysis using 3d Poisson equation – A quantitatively validated approach. *Medical Image Analysis*, 30:72 – 84, 2016.
- [17] M. Garland, A. Willmott, and P. S. Heckbert. Hierarchical face clustering on polygonal surfaces. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D ’01, pages 49–58, New York, NY, USA, 2001. ACM.
- [18] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP ’05. Eurographics Association, 2005.
- [19] M. Genctav, A. Genctav, and S. Tari. Nonlocal via local—nonlinear via linear: A new distance field via screened Poisson equation. *J. Math. Imaging Vis.*, 55(2):242–252, 2016.
- [20] G. Gilboa, J. Darbon, S. Osher, and T. Chan. Nonlocal convex functionals for image regularization. *UCLA CAM-report 06-57*, 2006.

- [21] D. Giorgi, S. Biasotti, and L. Paraboschi. Shape retrieval contest 2007: Water-tight models track. 8, 07 2008.
- [22] A. Golovinskiy and T. Funkhouser. Randomized cuts for 3d mesh analysis. *ACM Trans. Graph.*, 27(5):145:1–145:12, December 2008.
- [23] A. Golovinskiy and T. Funkhouser. Consistent segmentation of 3d models. *Computers & Graphics*, 33(3):262 – 269, 2009. {IEEE} International Conference on Shape Modelling and Applications 2009.
- [24] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.
- [25] A. Gregory, A. State, M. C. Lin, D. Manocha, and M. A. Livingston. Interactive surface decomposition for polyhedral morphing. *The Visual Computer*, 15(9):453–470, Dec 1999.
- [26] R. Guler, S. Tari, and G. Unal. Screened Poisson hyperfields for shape coding. *SIAM Journal on Imaging Sciences*, 7(4):2558–2590, 2014.
- [27] K. S. Gurumoorthy and A. Rangarajan. A schrödinger equation for the fast computation of approximate euclidean distance functions. In X.-C. Tai, K. Mørken, M. Lysaker, and K.-A. Lie, editors, *Scale Space and Variational Methods in Computer Vision: Second International Conference, SSVM 2009, Voss, Norway, June 1-5, 2009. Proceedings*, pages 100–111, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [28] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 203–212, New York, NY, USA, 2001. ACM.
- [29] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18(1-3):65–96, 1984.
- [30] D. D. Hoffman and M. Singh. Saliency of visual parts. *Cognition*, 63(1):29 – 78, 1997.

- [31] R. Hu, L. Fan, and L. Liu. Co-segmentation of 3d shapes via subspace clustering. *Computer Graphics Forum*, 31(5):1703–1713, 2012.
- [32] Q. Huang, V. Koltun, and L. Guibas. Joint shape segmentation with linear programming. *ACM Trans. Graph.*, 30(6):125:1–125:12, 2011.
- [33] M. Jung and L. Vese. Nonlocal variational image deblurring models in the presence of Gaussian or impulse noise. In *SSVM*, pages 401–412. Springer, 2009.
- [34] O. V. Kaick, N. Fish, Y. Kleiman, S. Asafi, and D. Cohen-OR. Shape segmentation by approximate convexity analysis. *ACM Trans. Graph.*, 34(1):4:1–4:11, December 2014.
- [35] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *ACM Trans. Graph.*, 29(4):102:1–102:12, 2010.
- [36] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception*. Praeger, 1979.
- [37] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, pages 279–286, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [38] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, 2005.
- [39] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph.*, 22(3):954–961, July 2003.
- [40] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP '06*, pages 61–70, 2006.
- [41] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin. Fast mesh segmentation using random walks. In *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling, SPM '08*, pages 183–191, New York, NY, USA, 2008. ACM.

- [42] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666, 2005.
- [43] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 362–371, New York, NY, USA, 2002. ACM.
- [44] X. Li, T. W. Woon, T. S. Tan, and Z. Huang. Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, pages 35–42, New York, NY, USA, 2001. ACM.
- [45] M. Limper, A. Kuijper, and D. W. Fellner. Mesh Saliency Analysis via Local Curvature Entropy. In T. Bashford-Rogers and L. P. Santos, editors, *EG 2016 - Short Papers*. The Eurographics Association, 2016.
- [46] H.-Y. S. Lin, H.-Y. Liao, and J.-C. Lin. Visual salience-guided mesh decomposition. *Multimedia, IEEE Transactions on*, 9(1):46–57, Jan 2007.
- [47] G. Liu, Y. Gingold, and J.-M. Lien. Continuous visibility feature. In *28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, June 2015. IEEE.
- [48] R. Liu and H. Zhang. Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum*, 26(3):385–394, 2007.
- [49] D. G. Lowe. Object recognition from local scale-invariant features. In *Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.
- [50] P. Luo, Z. Wu, C. Xia, L. Feng, and T. Ma. Co-segmentation of 3d shapes via multi-view spectral clustering. *The Visual Computer*, 29(6):587–597, 2013.
- [51] J. Lv, X. Chen, J. Huang, and H. Bao. Semi-supervised mesh segmentation and labeling. *Comput. Graph. Forum*, 31(7.2):2241–2248, September 2012.
- [52] A. Mangan and R. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):308–321, Oct 1999.

- [53] D. Marr and H. K. Nishihara. Representation and recognition of spatial organization of three dimensional shapes. In *Proceedings of the Royal Society of London, Series B, Biological Sciences*, 200, pages 269–294, 1978.
- [54] D. Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9(3):355–383, 1977.
- [55] D. Page, A. Koschan, and M. Abidi. Perception-based 3d triangle mesh segmentation using fast marching watersheds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–27–II–32 vol.2, June 2003.
- [56] A. Pasupathy and C. E. Connor. Population coding of shape in area V4. *Nature Neuroscience*, 5(2):1332–1338, 2002.
- [57] T. Peng, I. H. Jermyn, V. Prinet, and J. Zerubia. Extended phase field higher-order active contour models for networks. *Int. J. Computer Vision*, 88(1):111–128, 2010.
- [58] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.
- [59] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- [60] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.
- [61] S. Shlafman, A. Tal, and S. Katz. Metamorphosis of polyhedral surfaces using decomposition. *Computer Graphics Forum*, 21(3):219–228, 2002.
- [62] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. Graph.*, 30(6):126:1–126:10, December 2011.
- [63] J. Solomon, M. Ben-Chen, A. Butscher, and L. Guibas. Discovery of intrinsic primitives on triangle meshes. *Computer Graphics Forum*, 30(2):365–374, 2011.

- [64] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin. Mesh saliency via spectral processing. *ACM Trans. Graph.*, 33(1):6:1–6:17, 2014.
- [65] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009.
- [66] S. Tari. Hierarchical shape decomposition via level sets. In M. Wilkinson and J. Roerdink, editors, *Mathematical Morphology and Its Application to Signal and Image Processing*, volume 5720 of *Lecture Notes in Computer Science*, pages 215–225. Springer Berlin Heidelberg, 2009.
- [67] S. Tari. Extracting parts of 2d shapes using local and global interactions simultaneously. In C. Chen, editor, *Handbook of Pattern Recognition and Computer Vision*. World Scientific, Jan 2010.
- [68] S. Tari and M. Genctav. From a non-local ambrosio-tortorelli phase field to a randomized part hierarchy tree. *Journal of Mathematical Imaging and Vision*, 49(1):69–86, 2014.
- [69] S. Tari, J. Shah, and H. Pien. A computationally efficient shape analysis via level sets. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 1996.
- [70] Z. S. G. Tari, J. Shah, and H. Pien. Extraction of shape skeletons from grayscale images. *Computer Vision and Image Understanding*, 66(2):133–146, 1997.
- [71] P. Theologou, I. Pratikakis, and T. Theoharis. Unsupervised spectral mesh segmentation driven by heterogeneous graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):397–410, Feb 2017.
- [72] M. Toutain, A. Elmoataz, F. Lozes, and A. Mansouri. Non-local discrete  $\infty$ -Poisson and Hamilton Jacobi equations. *J. Math. Imaging Vis.*, 55(2):229–241, June 2016.
- [73] H. Wang, T. Lu, O. K.-C. Au, and C.-L. Tai. Spectral 3D mesh segmentation with a novel single segmentation field. *Graphical Models*, 76(5):440–456, 2014.

- [74] D. Xu, H. Zhang, Q. Wang, and H. Bao. Poisson shape interpolation. *Graphical Models*, 68(3):268–281, 2006.
- [75] K. Xu, H. Li, H. Zhang, D. Cohen-Or, Y. Xiong, and Z.-Q. Cheng. Style-content separation by anisotropic part scales. In *ACM SIGGRAPH Asia 2010 Papers*, SIGGRAPH ASIA '10, pages 184:1–184:10, New York, NY, USA, 2010. ACM.
- [76] Y.-L. Yang, Y.-K. Lai, S.-M. Hu, and H. Pottmann. Robust principal curvatures on multiple scales. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 223–226, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [77] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with Poisson-based gradient field manipulation. *ACM Trans. Graph.*, 23(3):644–651, 2004.
- [78] J. Zhang, J. Zheng, C. Wu, and J. Cai. Variational mesh decomposition. *ACM Trans. Graph.*, 31(3):21:1–21:14, 2012.
- [79] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.
- [80] M. Zöckler, D. Stalling, and H.-C. Hege. Fast and intuitive generation of geometric shape transitions. *The Visual Computer*, 16(5):241–253, Jun 2000.
- [81] E. Zuckerberger, A. Tal, and S. Shlafman. Polyhedral surface decomposition with applications. *Computers & Graphics*, 26(5):733 – 743, 2002.



## **APPENDIX A**

### **SEGMENTATION PERFORMANCE ON PRINCETON BENCHMARK SET**

In this appendix, we show representative segmentations that form the basis for the benchmark performance of the proposed unsupervised mesh segmentation algorithm given in Chapter 6.

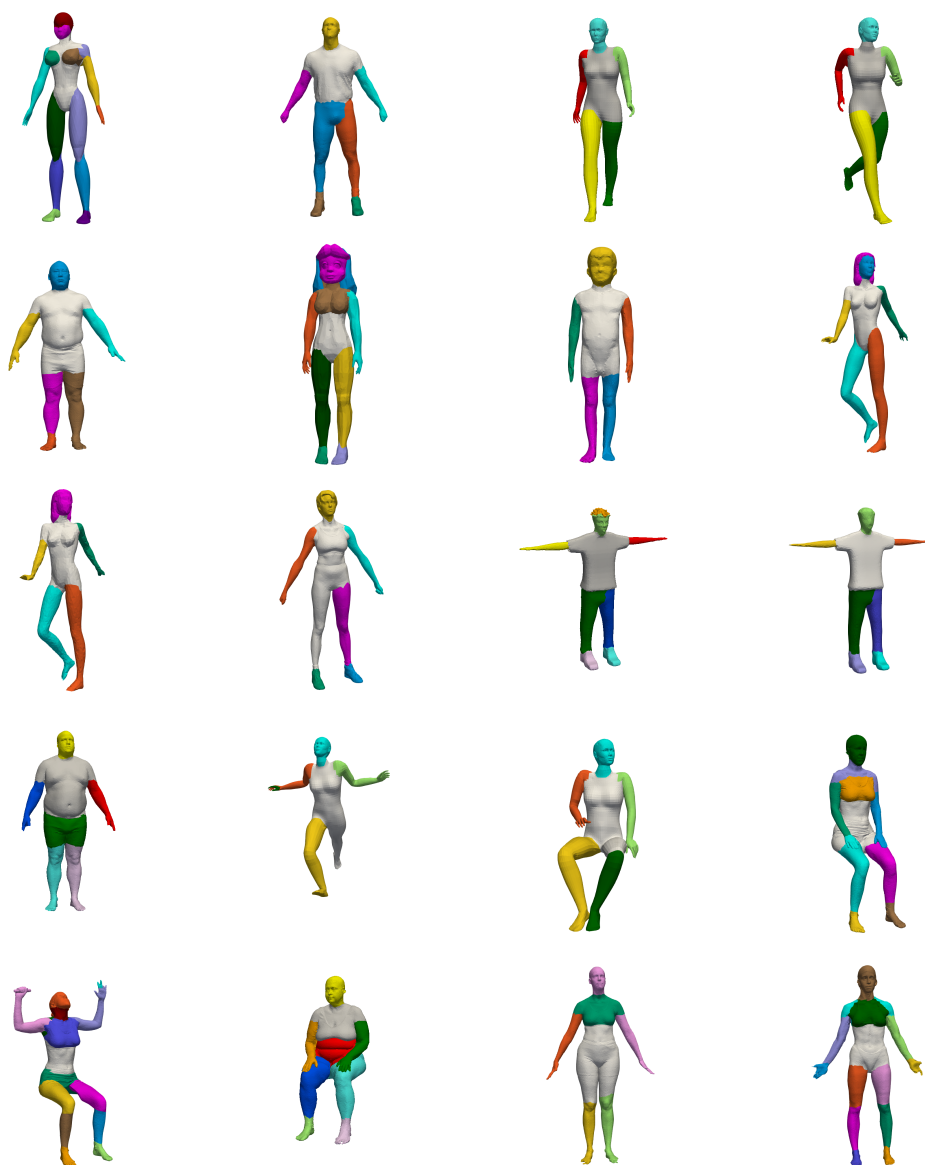


Figure A.1: Segmentation results for HUMAN category.



Figure A.2: Segmentation results for CUP category.

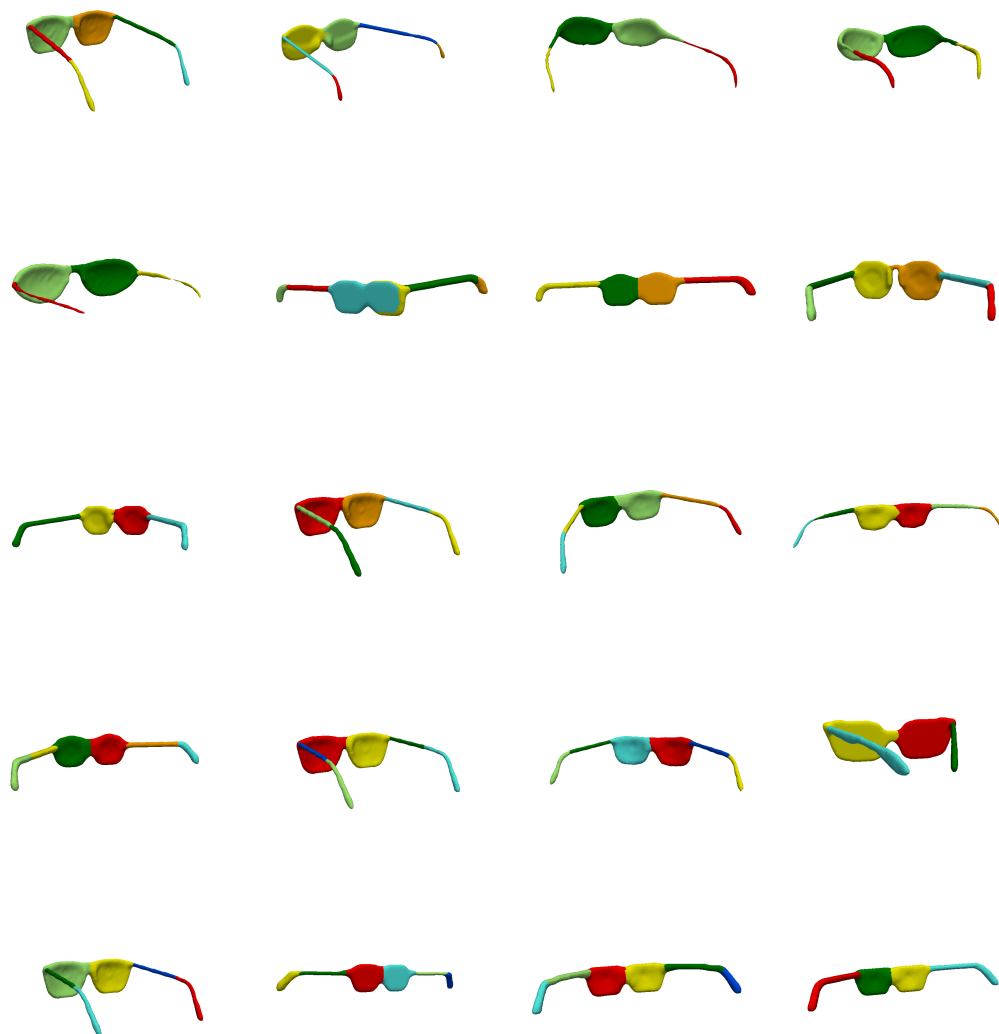


Figure A.3: Segmentation results for GLASSES category.

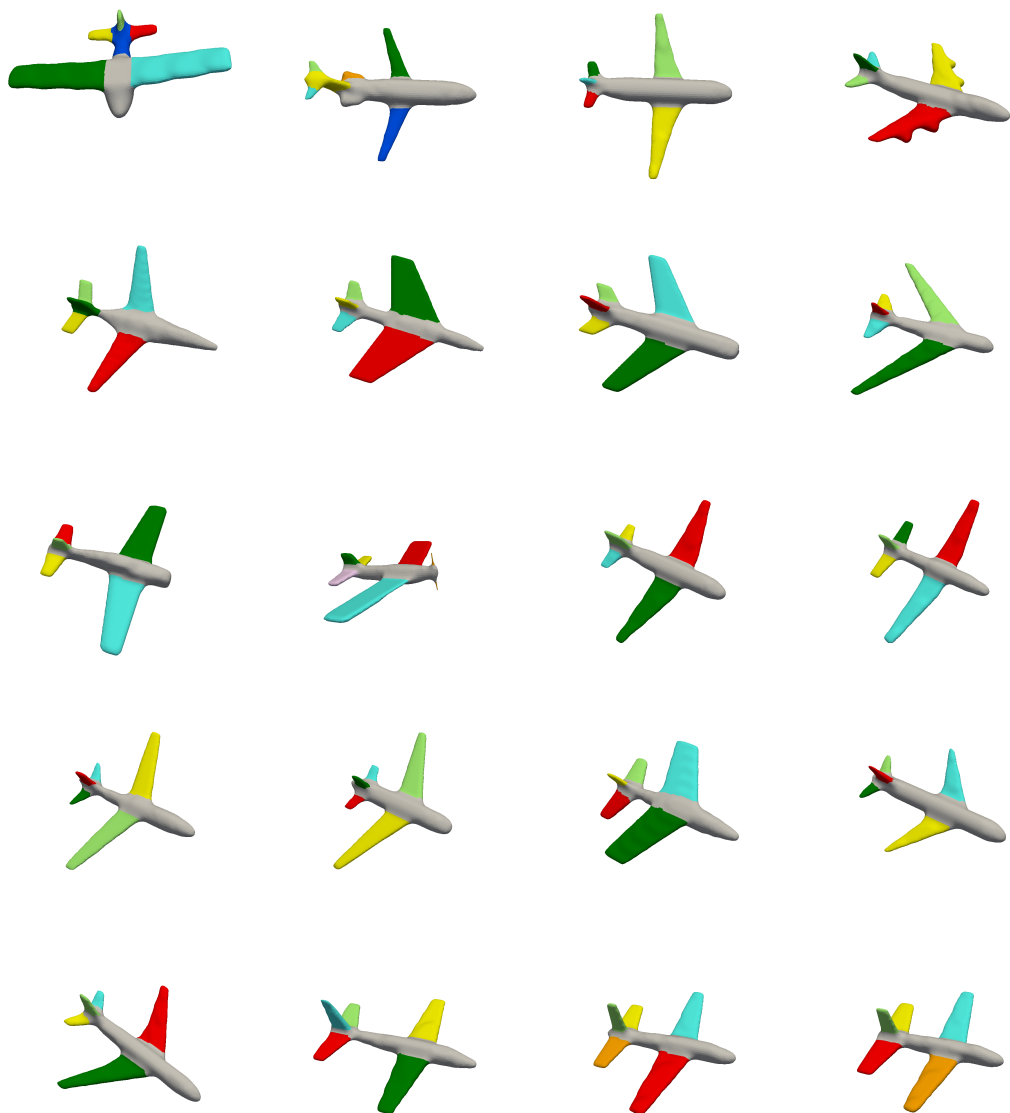


Figure A.4: Segmentation results for AIRPLANE category.

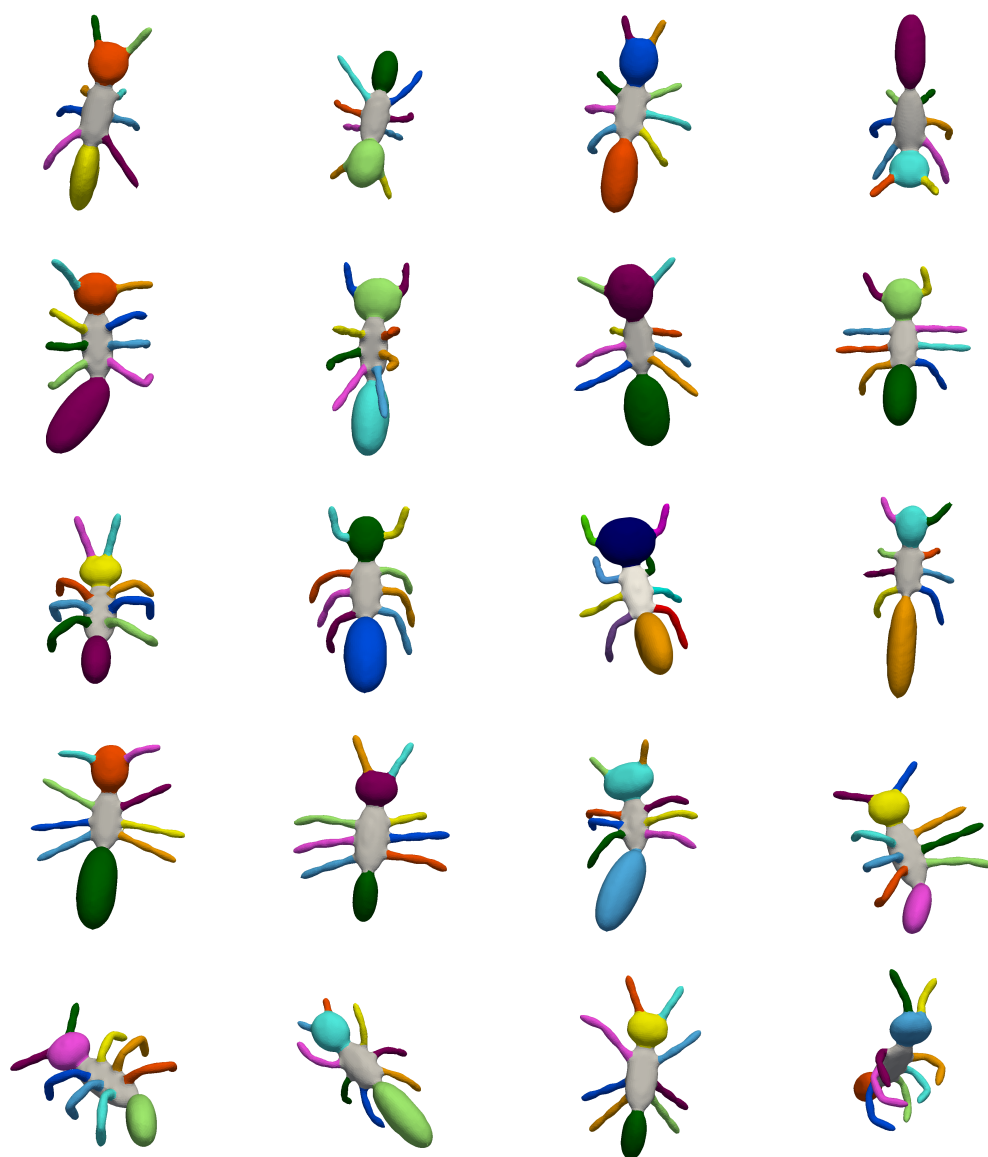


Figure A.5: Segmentation results for ANT category.



Figure A.6: Segmentation results for CHAIR category.

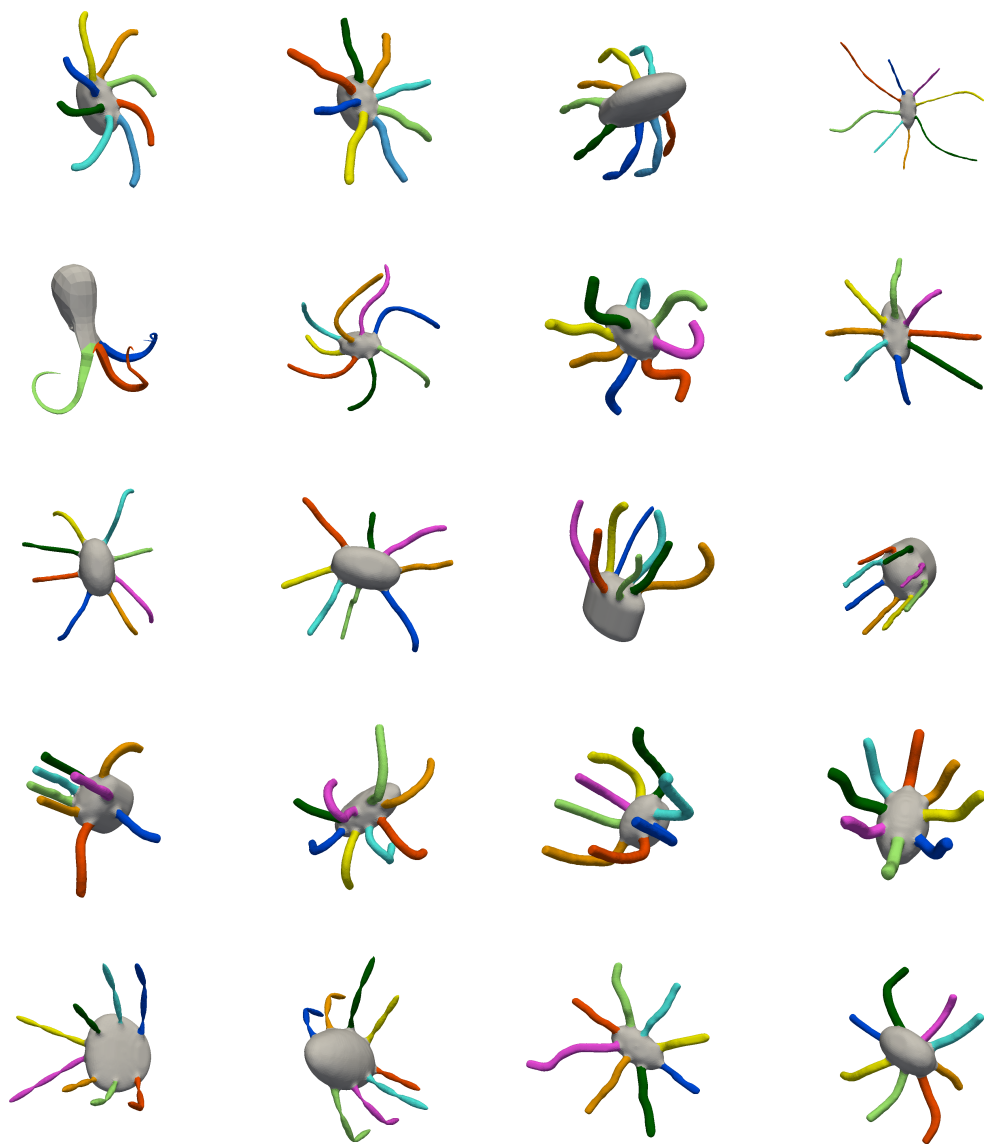


Figure A.7: Segmentation results for OCTOPUS category.



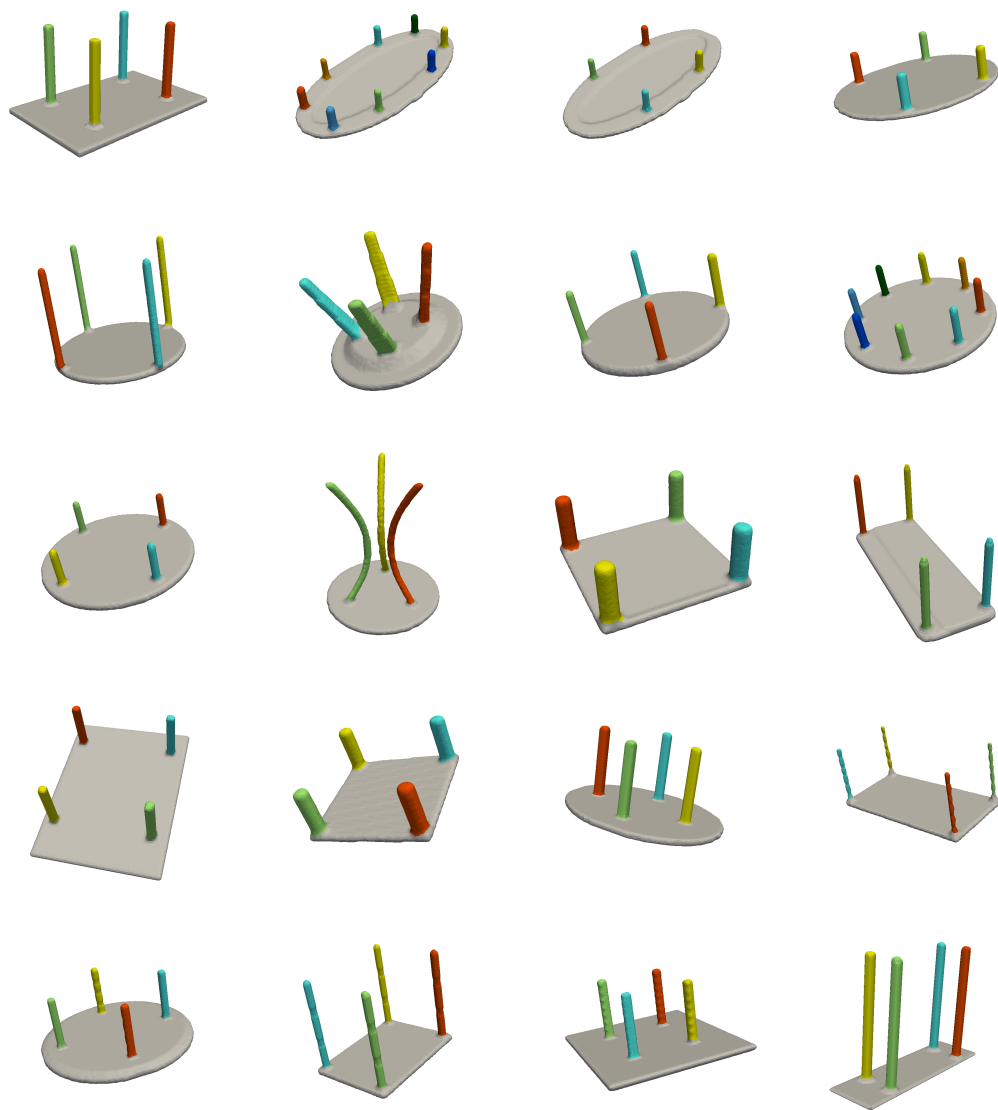


Figure A.8: Segmentation results for TABLE category.

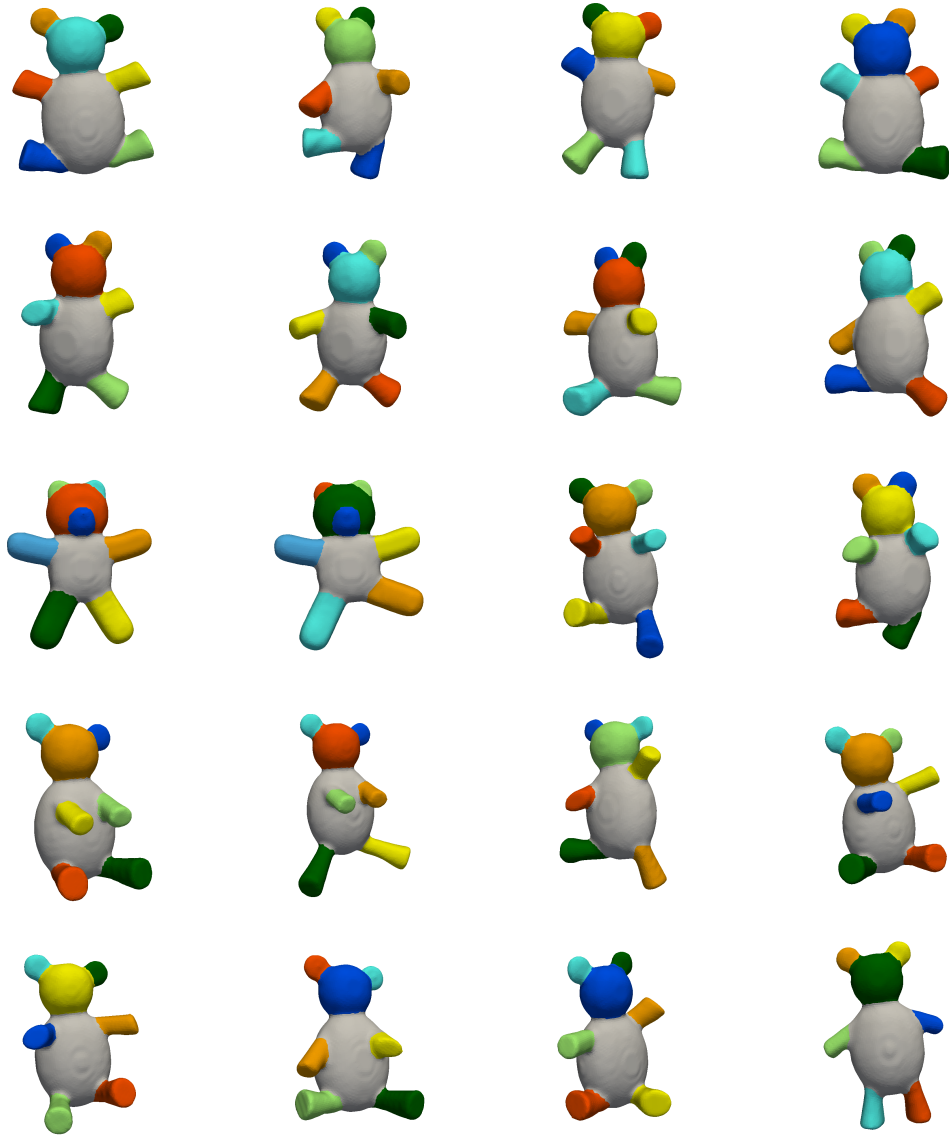


Figure A.9: Segmentation results for TEDDY category.

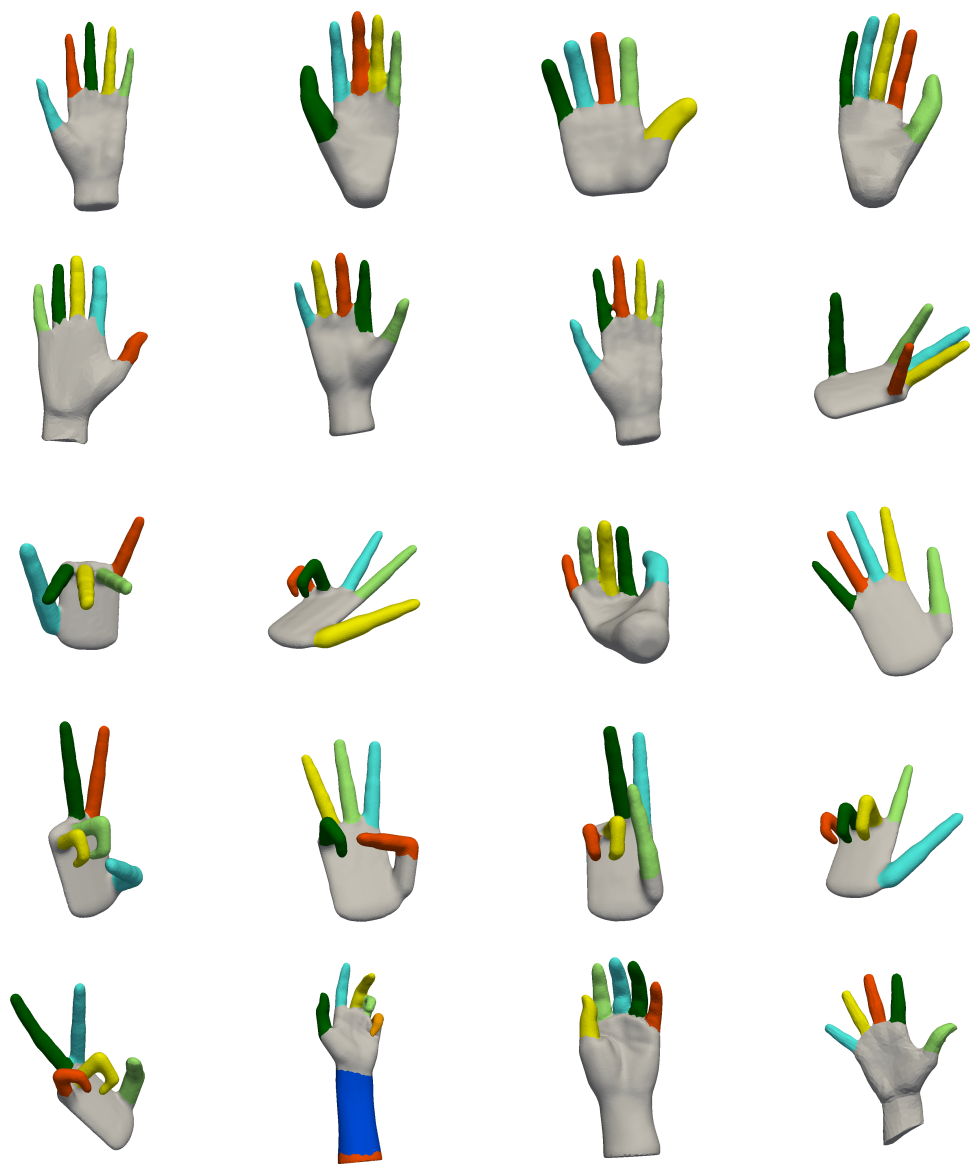


Figure A.10: Segmentation results for HAND category.

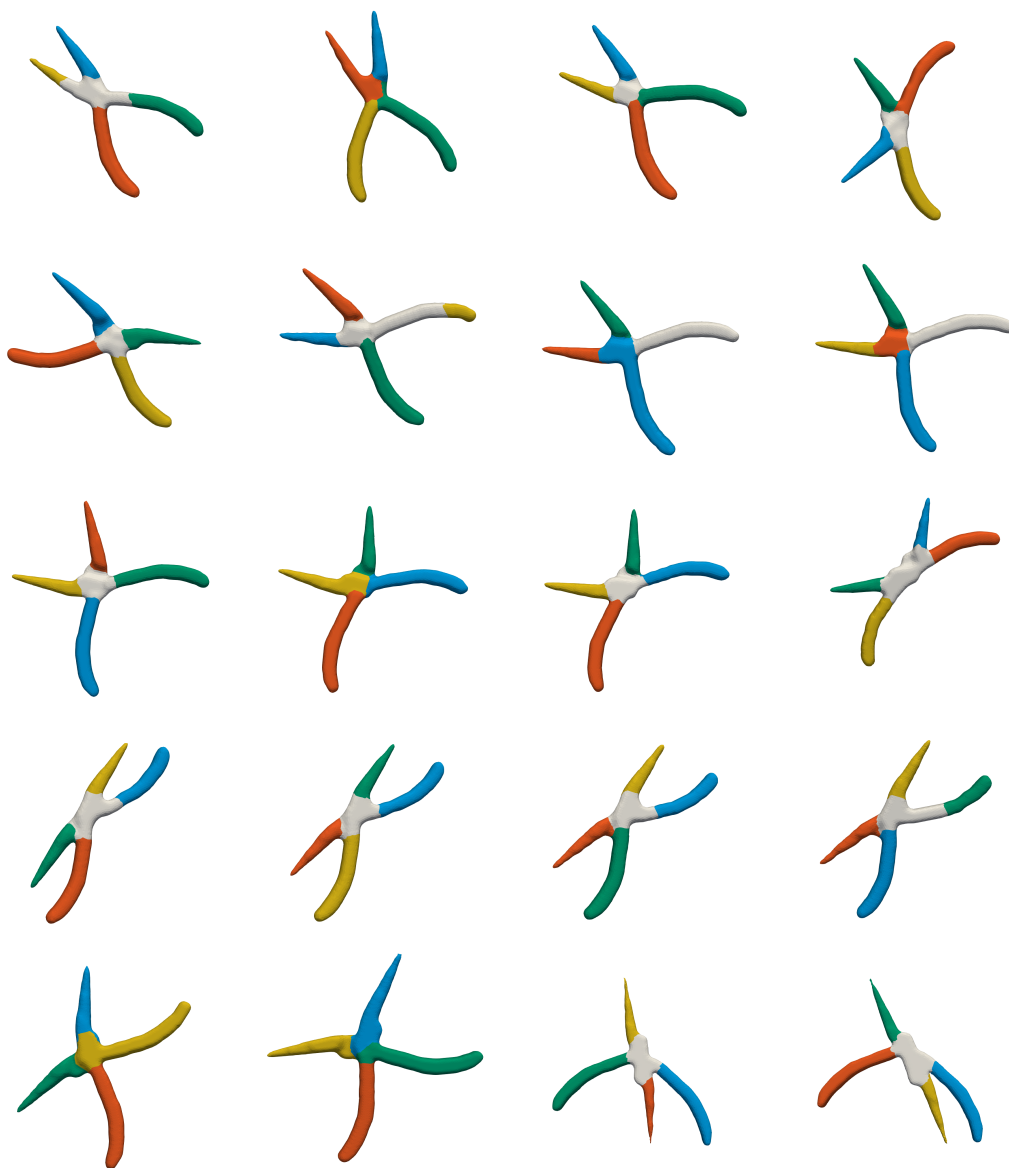


Figure A.11: Segmentation results for PLIER category.

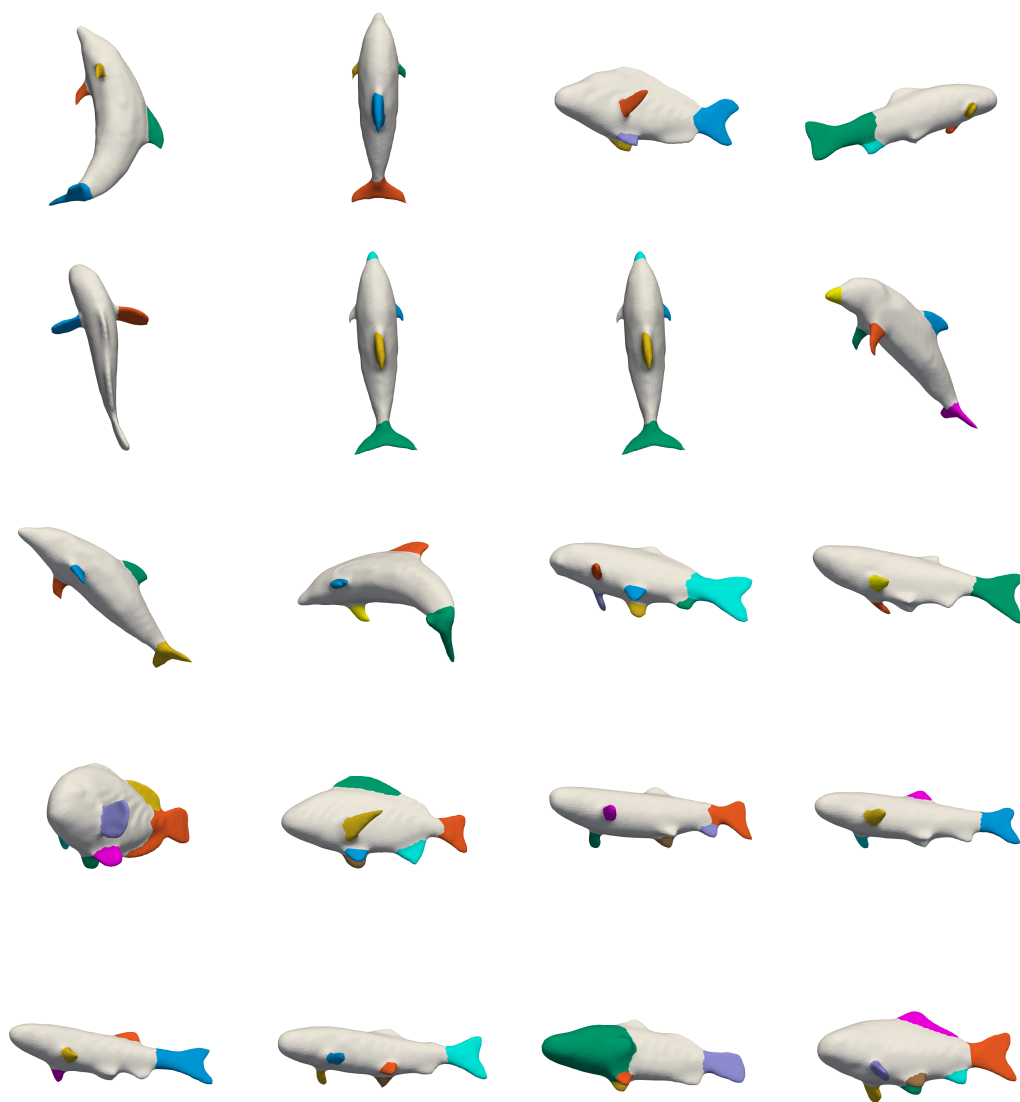


Figure A.12: Segmentation results for FISH category.

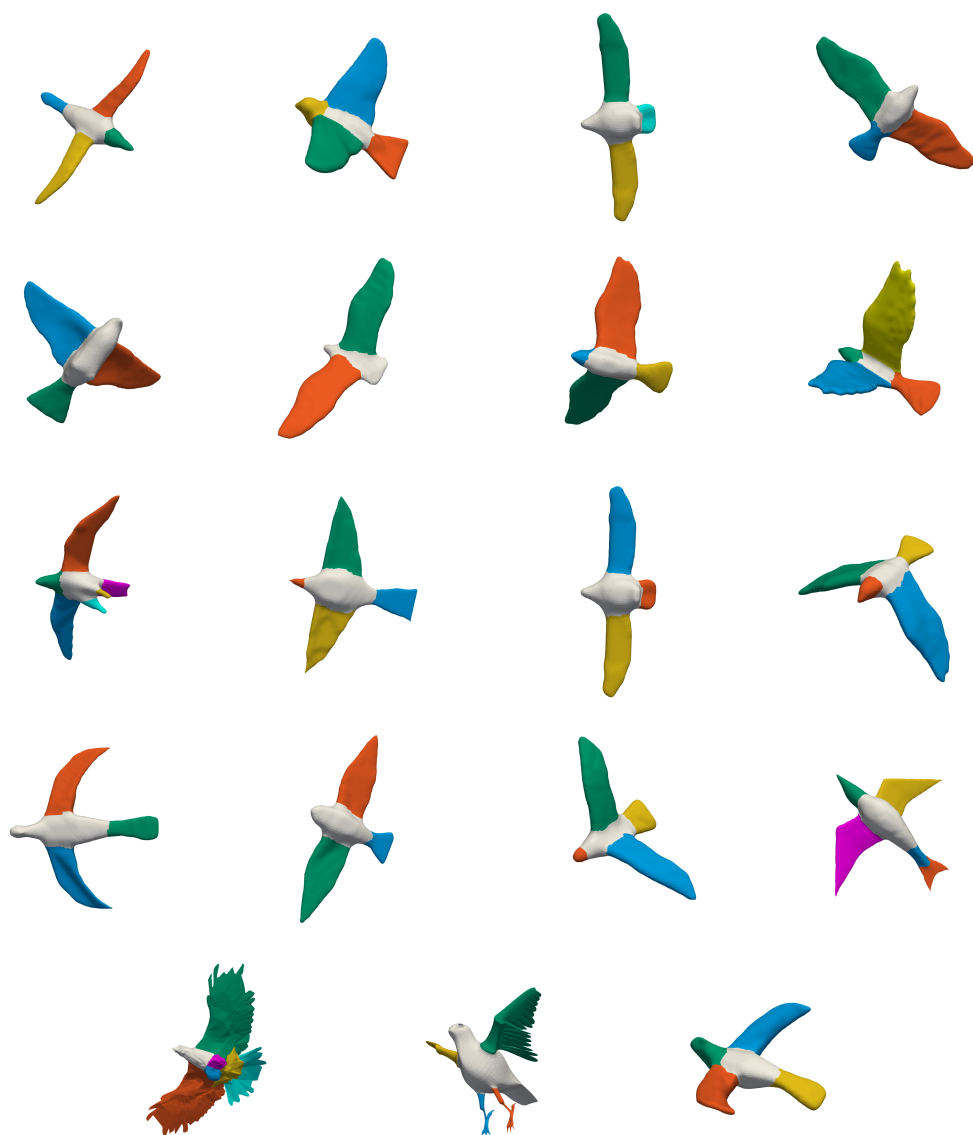


Figure A.13: Segmentation results for BIRD category.



Figure A.14: Segmentation results for ARMADILLO category.



Figure A.15: Segmentation results for BUST category.



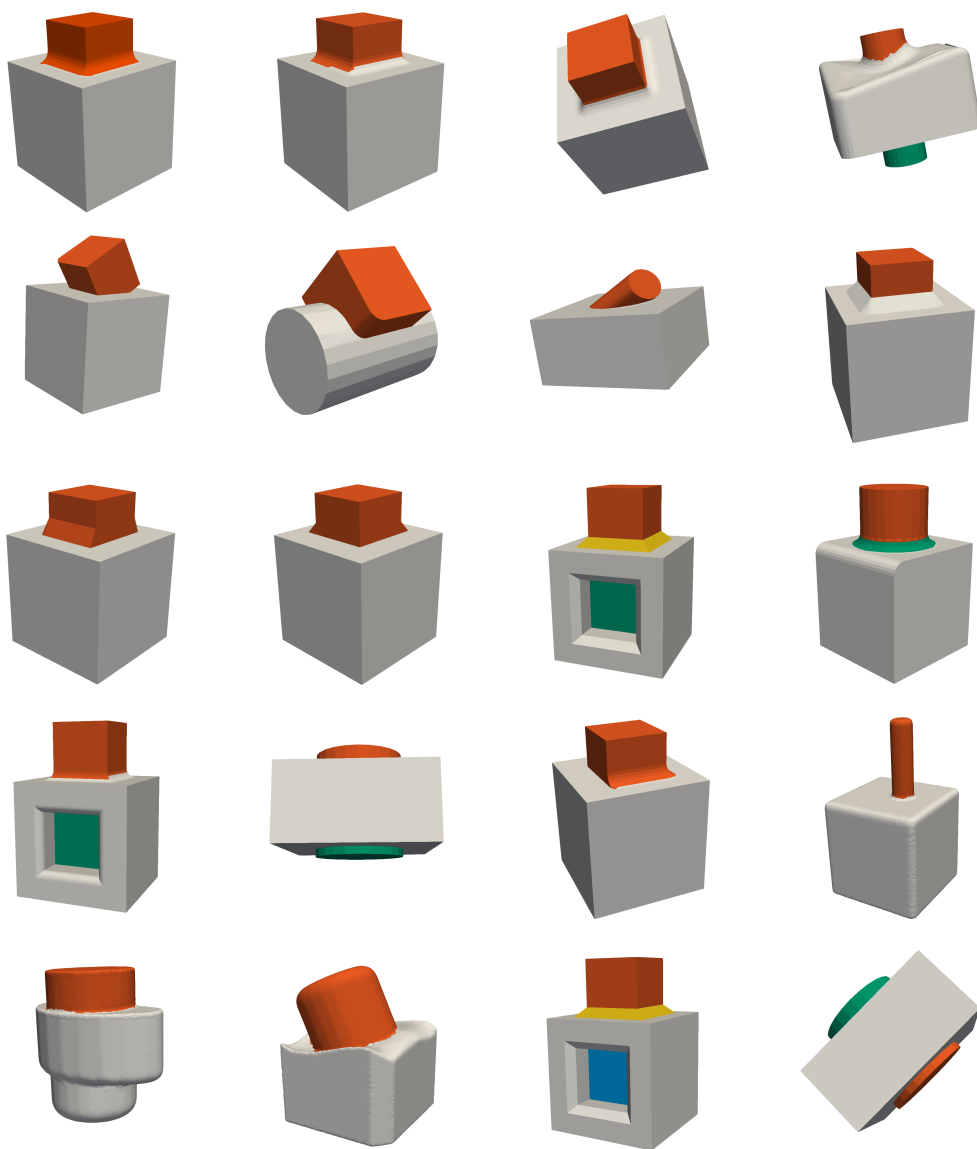


Figure A.16: Segmentation results for MECH category.

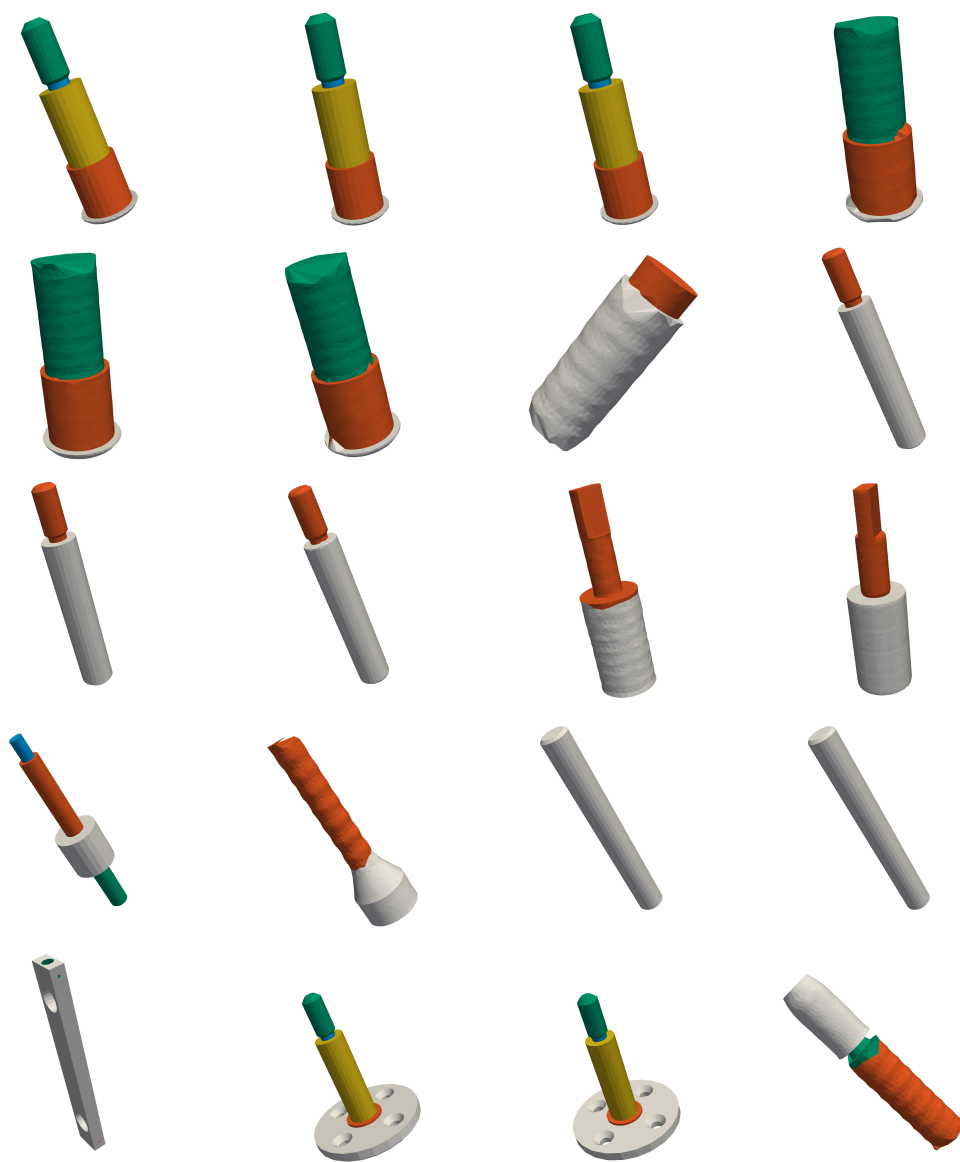


Figure A.17: Segmentation results for BEARING category.



Figure A.18: Segmentation results for VASE category.



Figure A.19: Segmentation results for FOURLEG category.

## **APPENDIX B**

### **GROUPWISE ANALYSIS RESULTS WITH 1000-SHAPES DATASET**

In this appendix, we demonstrate the performance of our groupwise analysis method on 1000-shapes dataset. In order to compactly present the decomposition hierarchy trees, we superimpose tree-nodes, i.e. draw the part stored in a tree node on top of the part stored in its parent node. Moreover, we coded different levels of the decomposition hierarchy using different colors. The shape itself (root) is drawn with light-gray. The parts that are split in the first level are shown in blue color. And finally, the second level is coded using orange. An example to construction of superimposed representation is shown in Figure B.1.

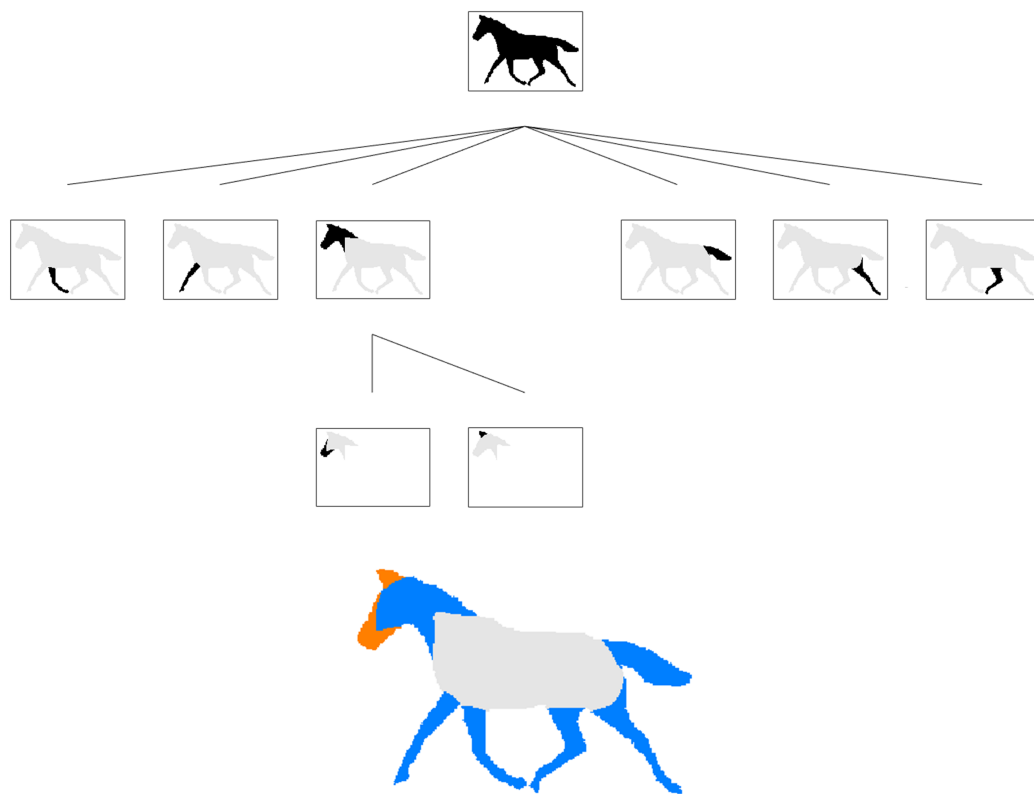


Figure B.1: From decomposition hierarchy tree representation to compact representation with superimposed nodes. The shape itself (root) is drawn with light-gray. The parts that are split in the first level are shown in blue color, whereas the second level is coded using orange.

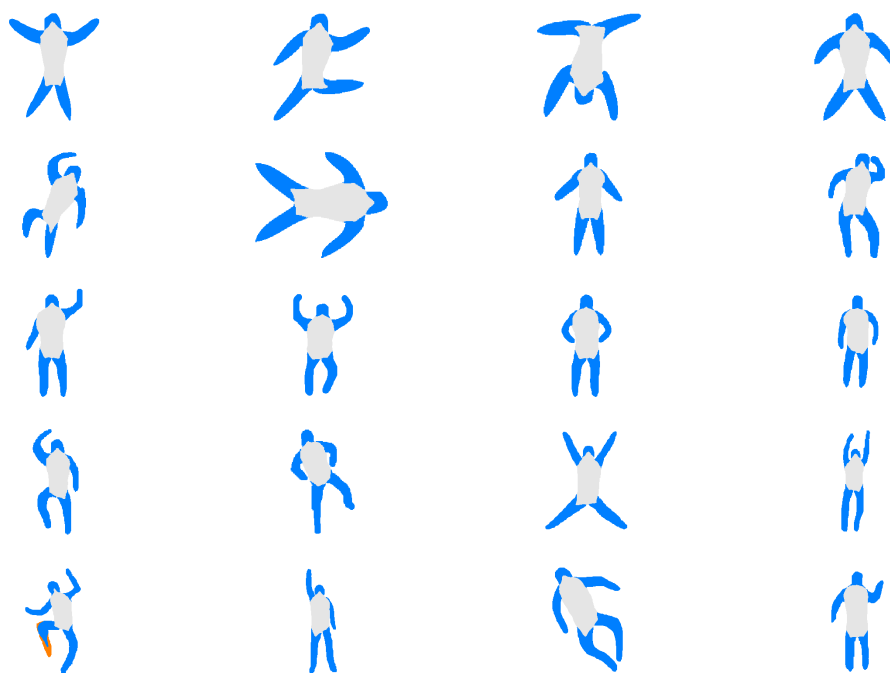


Figure B.2: Groupwise analysis results for category 1.

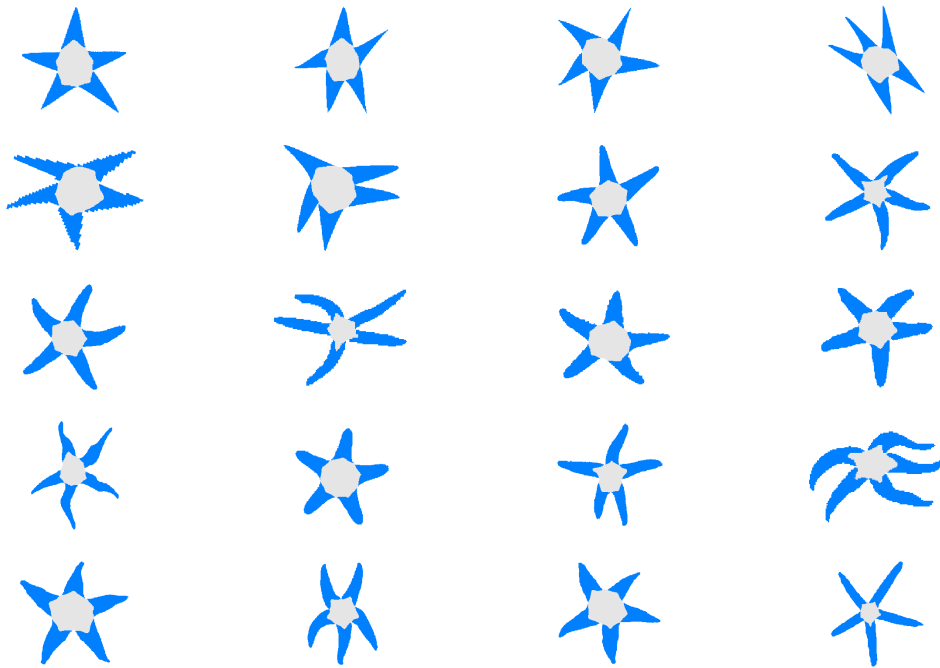


Figure B.3: Groupwise analysis results for category 2.



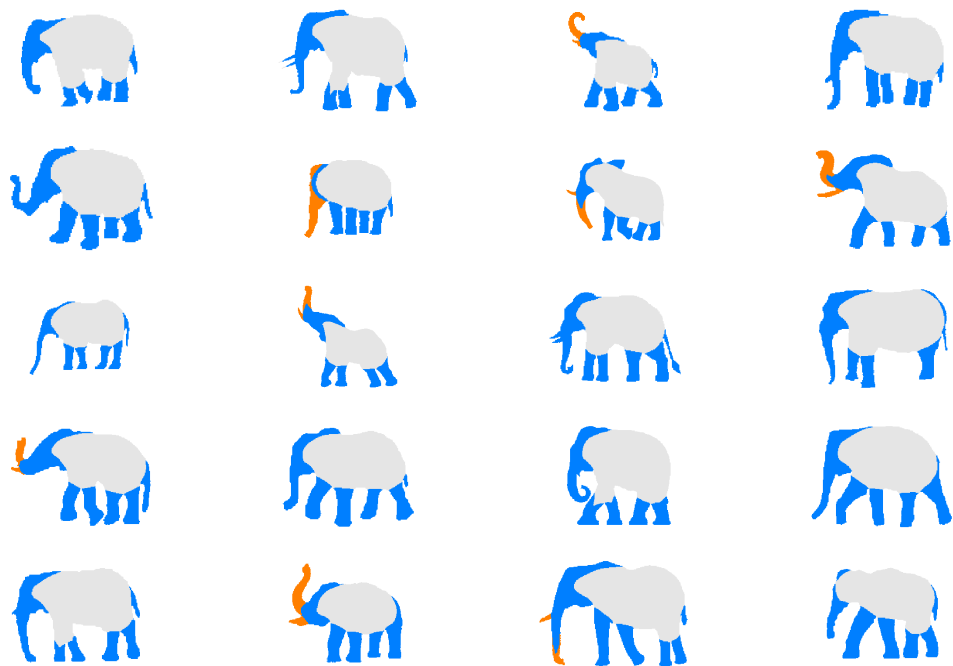


Figure B.4: Groupwise analysis results for category 3.

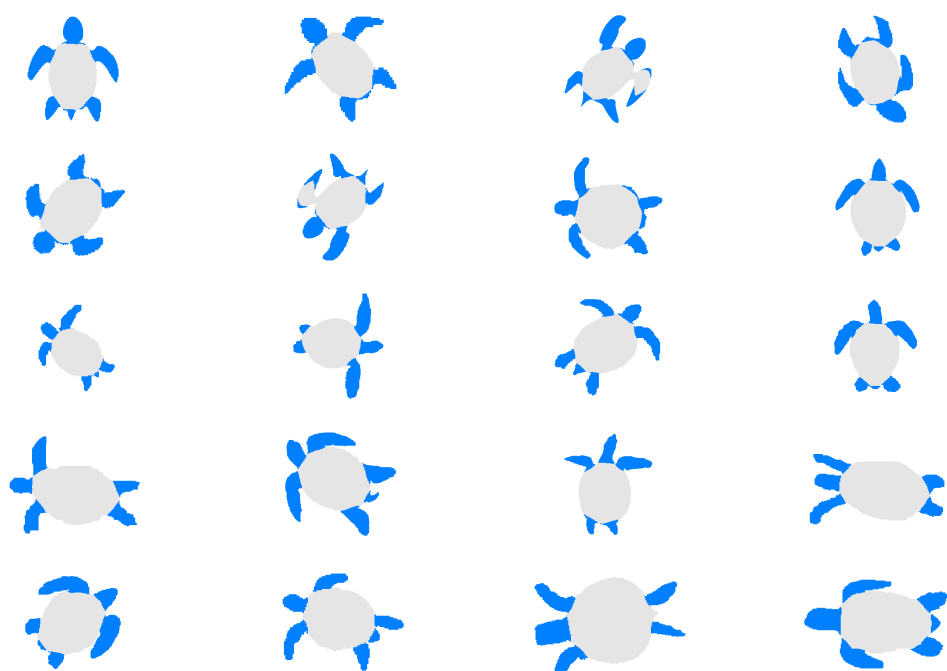


Figure B.5: Groupwise analysis results for category 4.

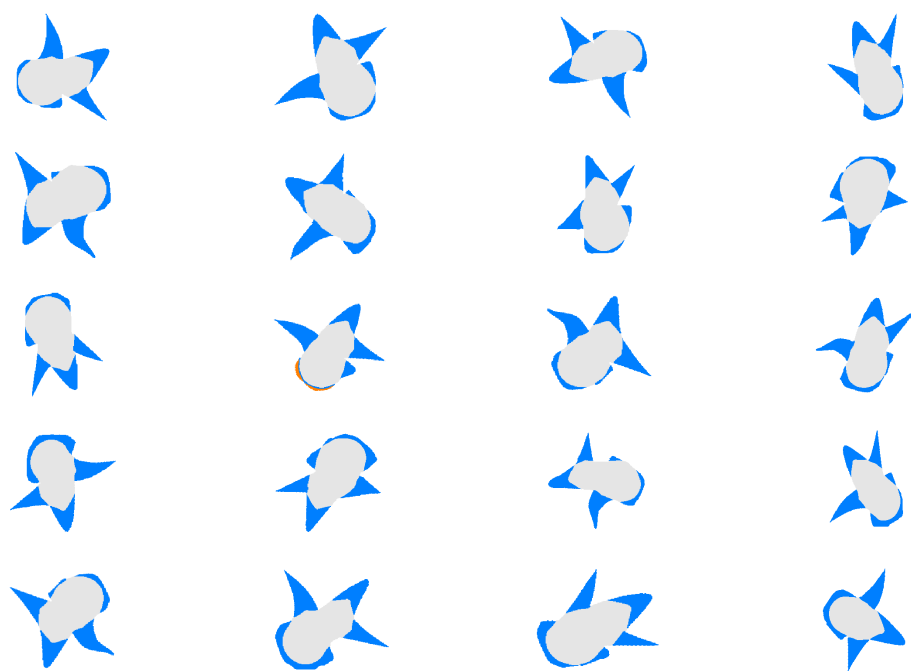


Figure B.6: Groupwise analysis results for category 5.

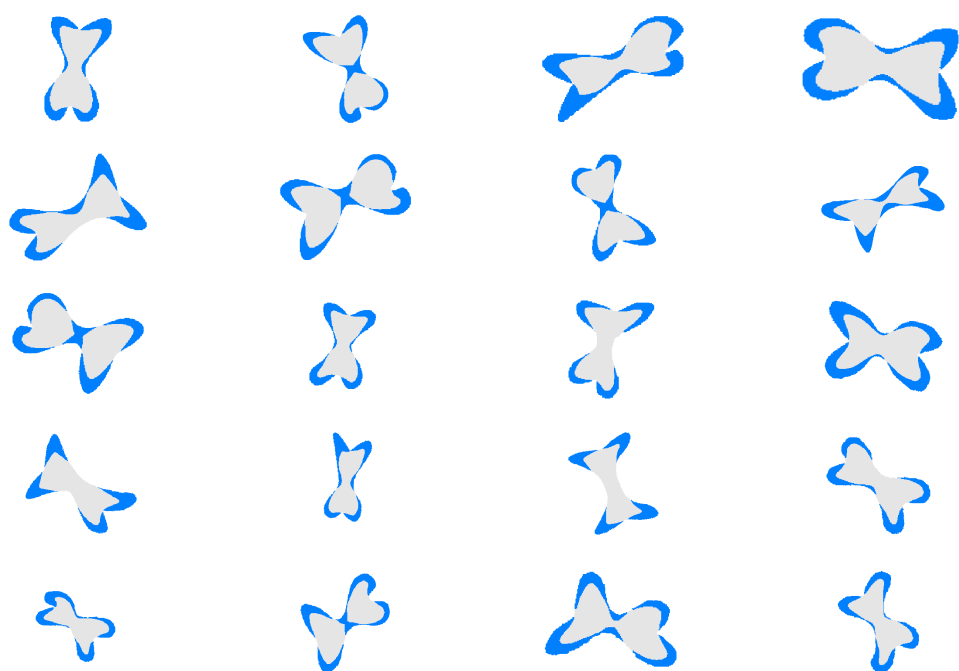


Figure B.7: Groupwise analysis results for category 6.

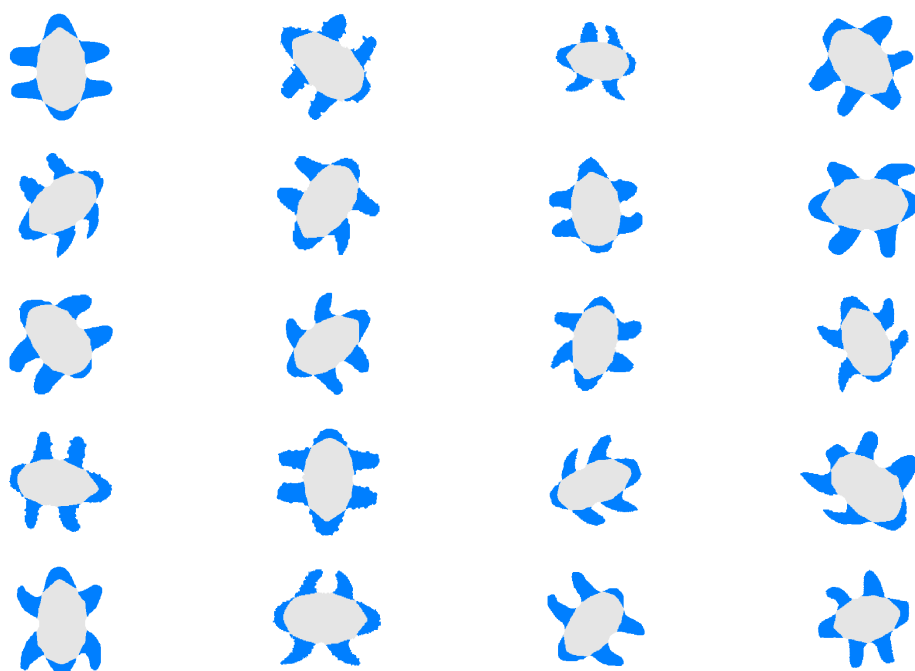


Figure B.8: Groupwise analysis results for category 7.

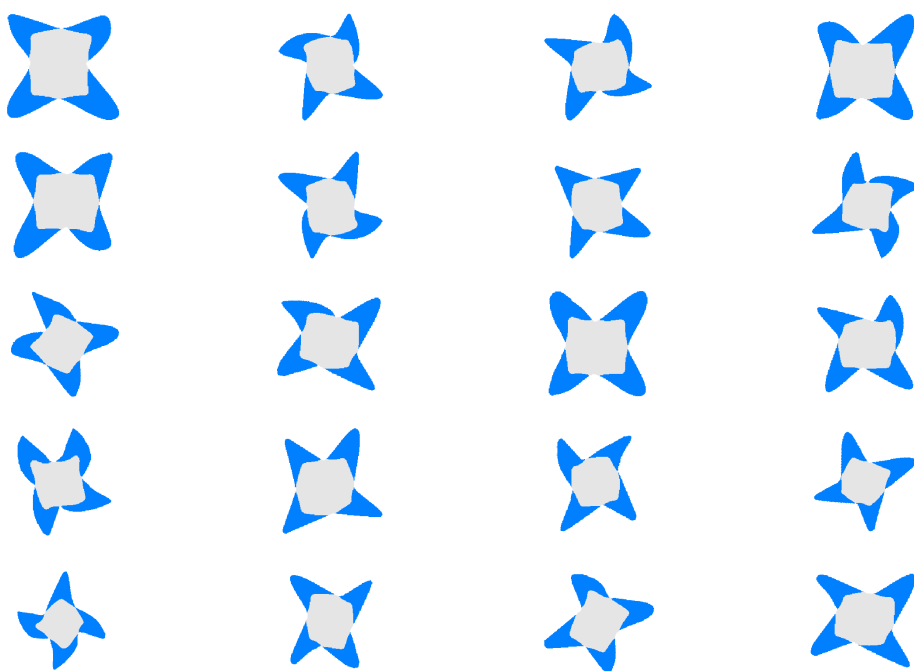


Figure B.9: Groupwise analysis results for category 8.



Figure B.10: Groupwise analysis results for category 9.

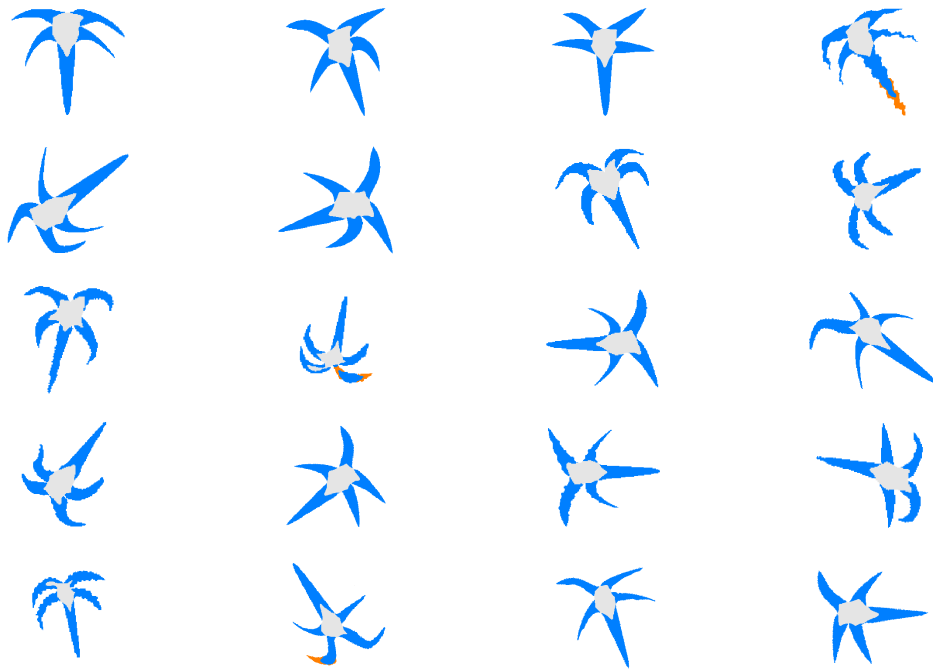


Figure B.11: Groupwise analysis results for category 10.



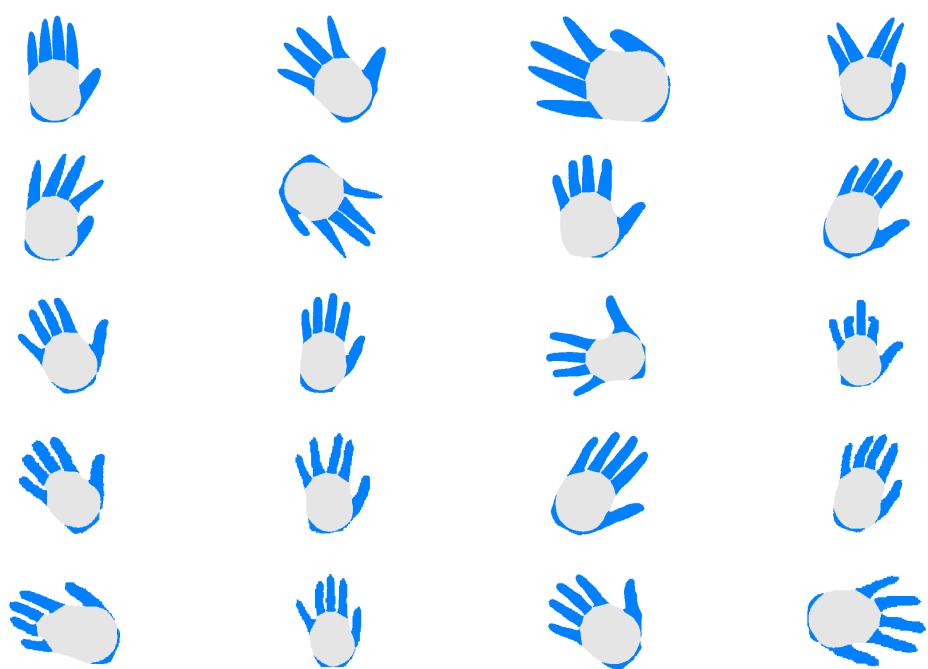


Figure B.12: Groupwise analysis results for category 11.

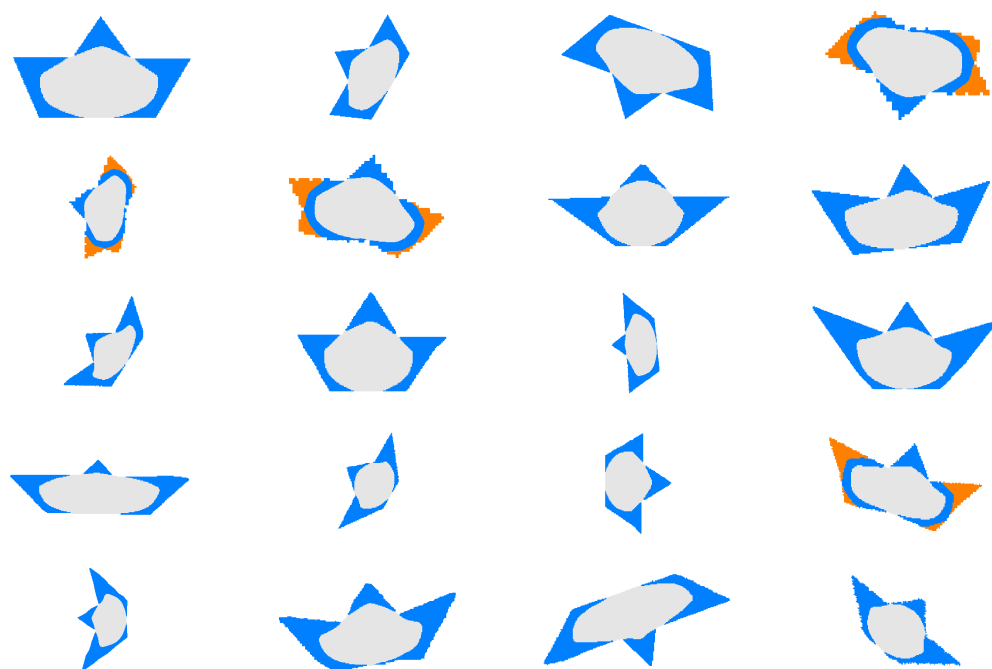


Figure B.13: Groupwise analysis results for category 12.

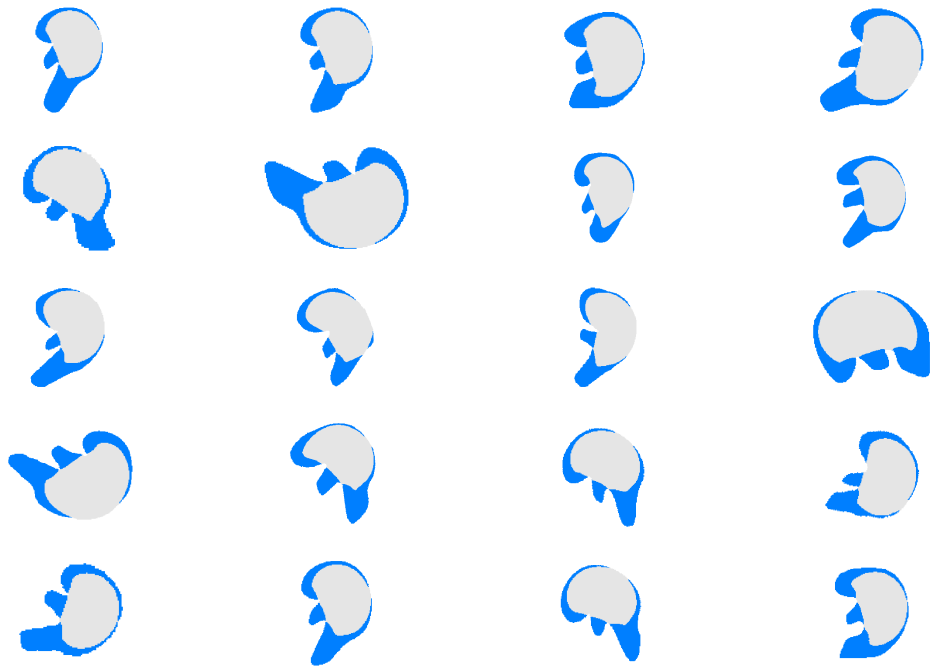


Figure B.14: Groupwise analysis results for category 13.

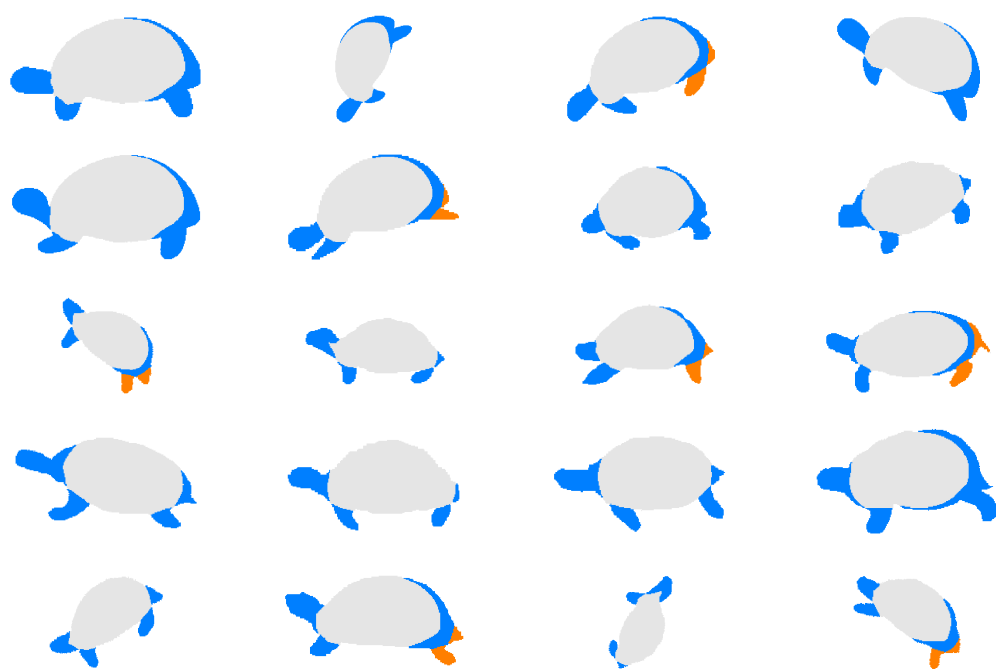


Figure B.15: Groupwise analysis results for category 14.

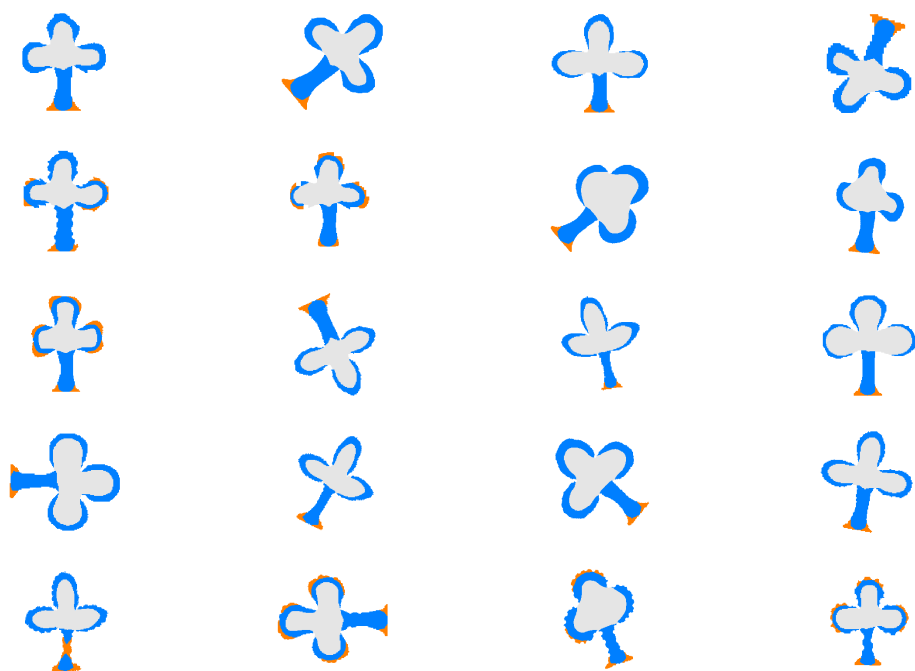


Figure B.16: Groupwise analysis results for category 15.

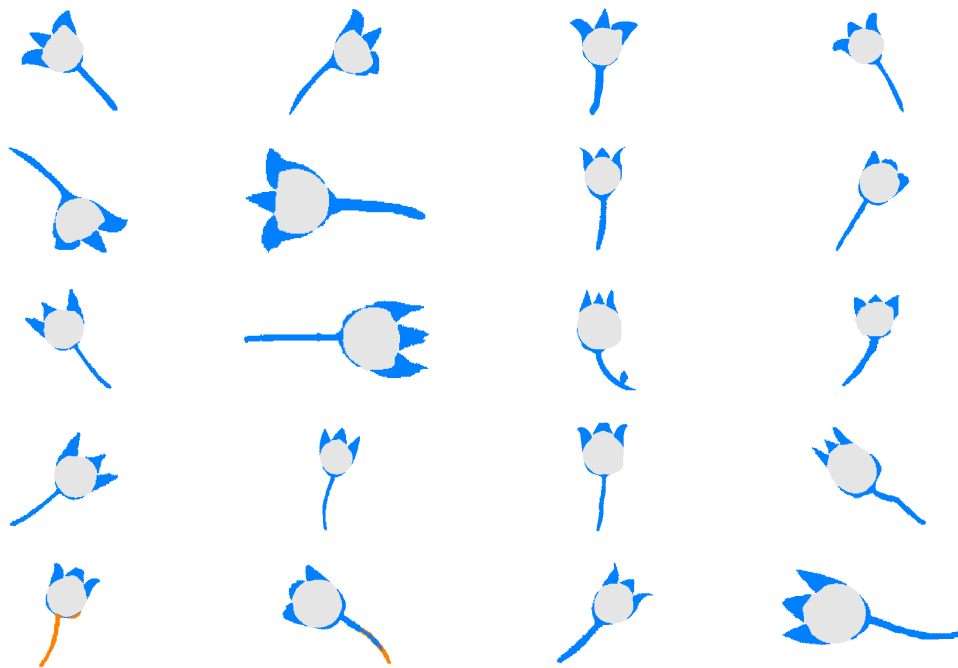


Figure B.17: Groupwise analysis results for category 16.

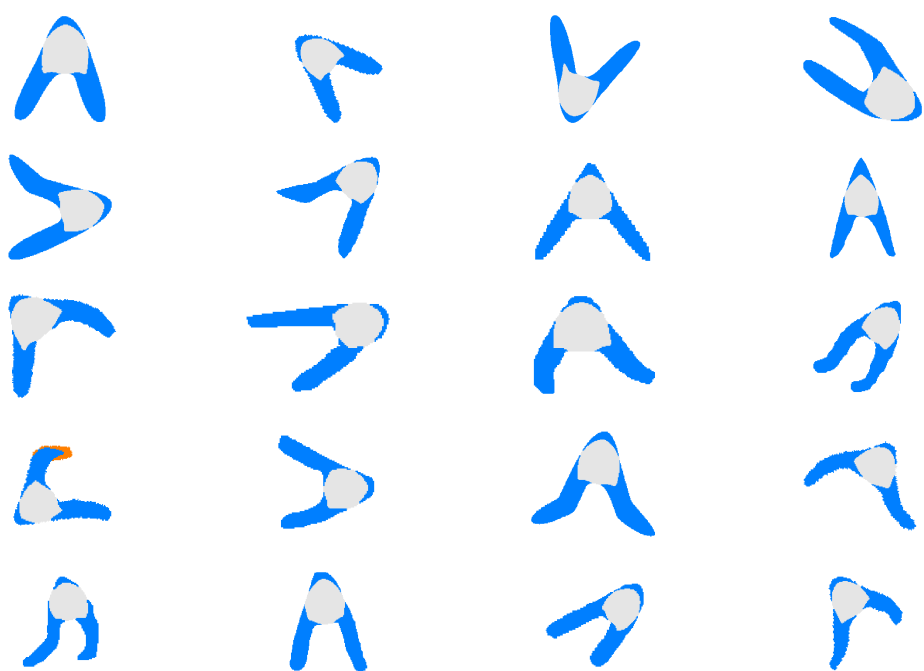


Figure B.18: Groupwise analysis results for category 17.



Figure B.19: Groupwise analysis results for category 18.



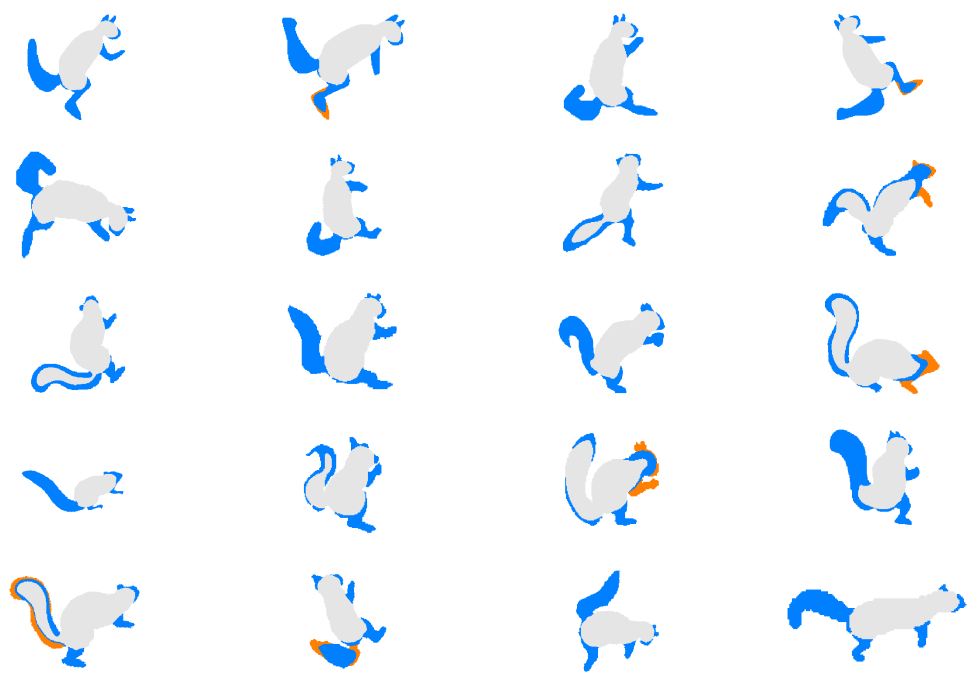


Figure B.20: Groupwise analysis results for category 19.

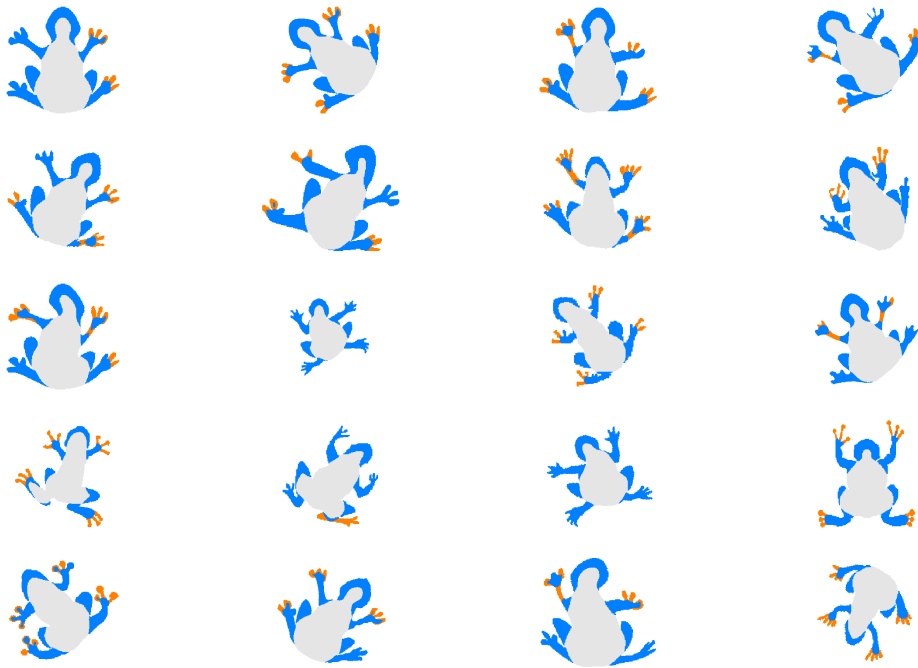


Figure B.21: Groupwise analysis results for category 20.

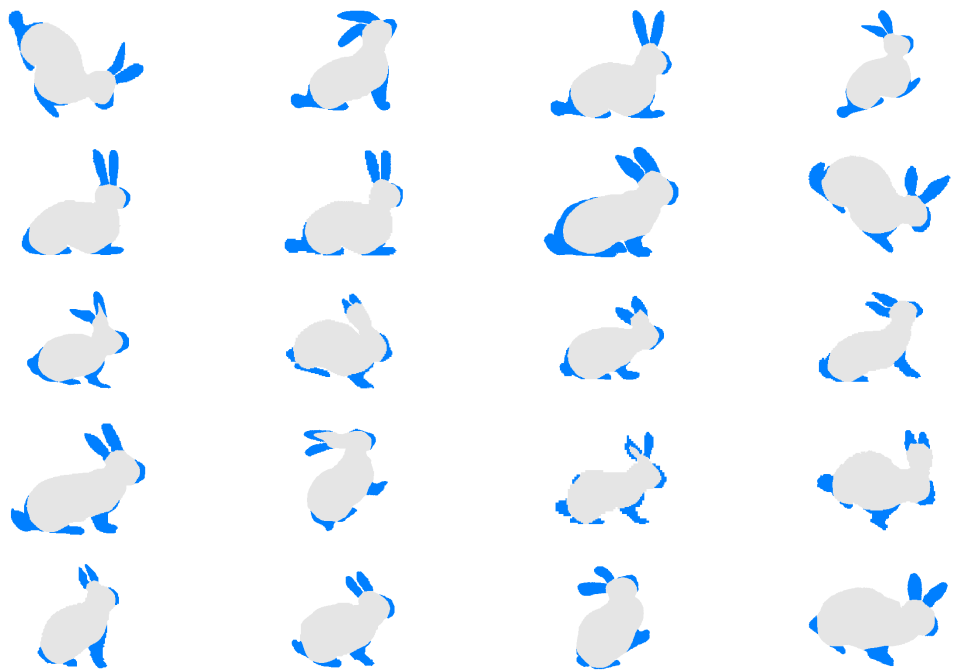


Figure B.22: Groupwise analysis results for category 21.

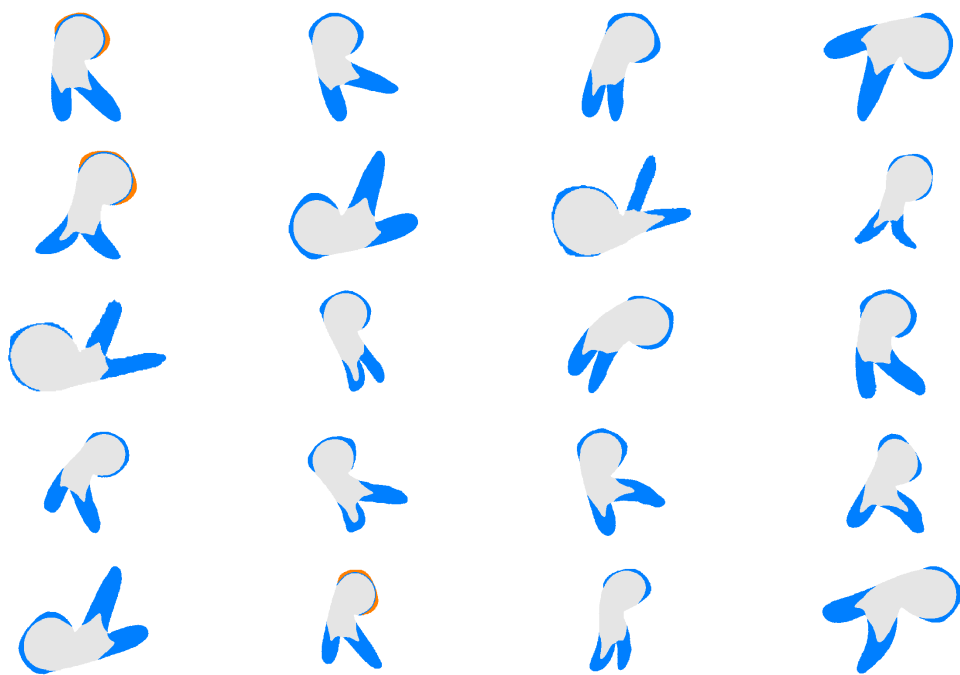


Figure B.23: Groupwise analysis results for category 22.

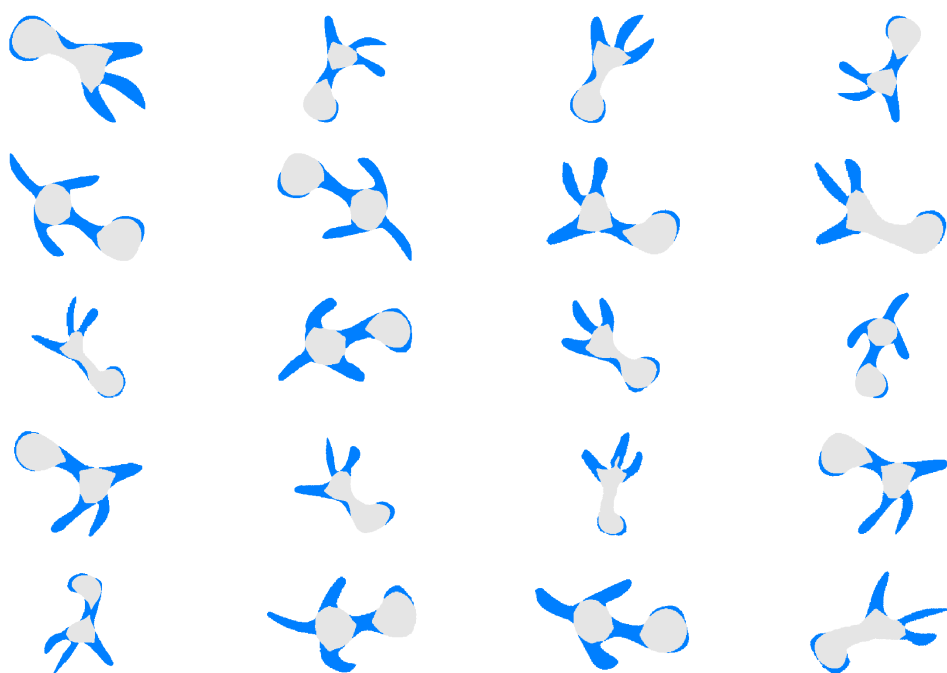


Figure B.24: Groupwise analysis results for category 23.



Figure B.25: Groupwise analysis results for category 24.

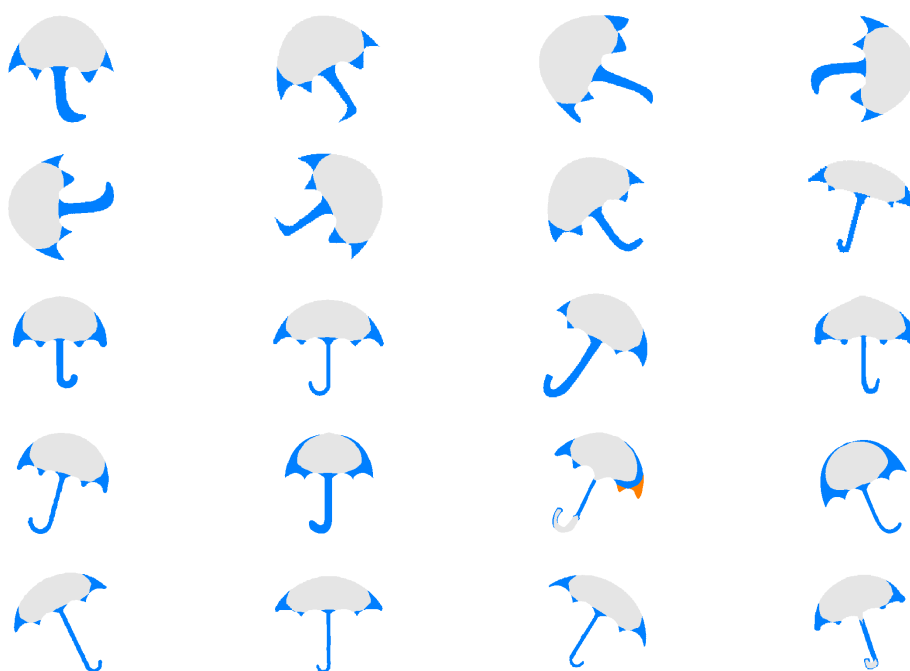


Figure B.26: Groupwise analysis results for category 25.

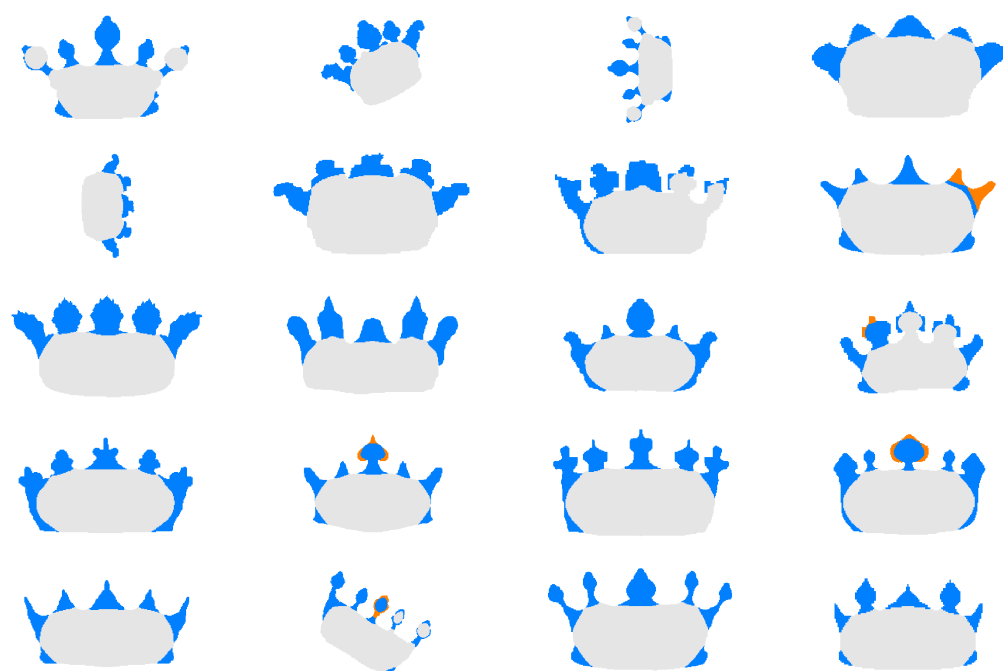


Figure B.27: Groupwise analysis results for category 26.



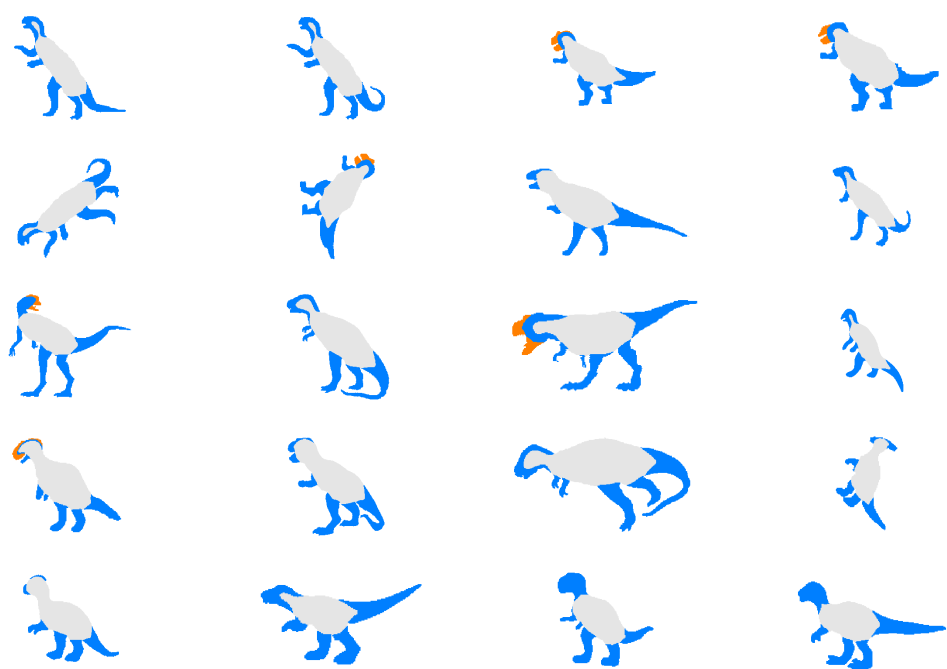


Figure B.28: Groupwise analysis results for category 27.

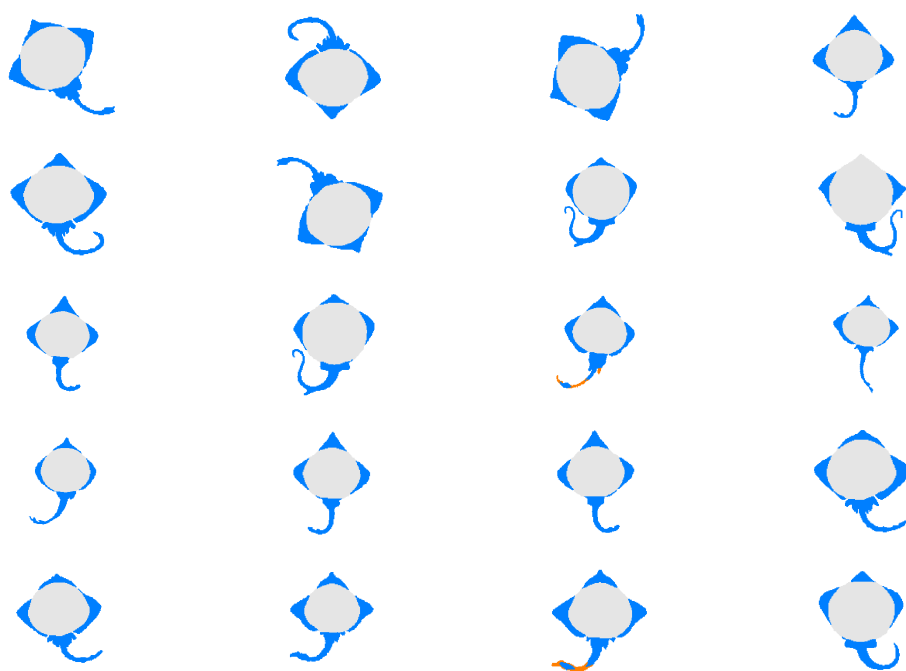


Figure B.29: Groupwise analysis results for category 28.

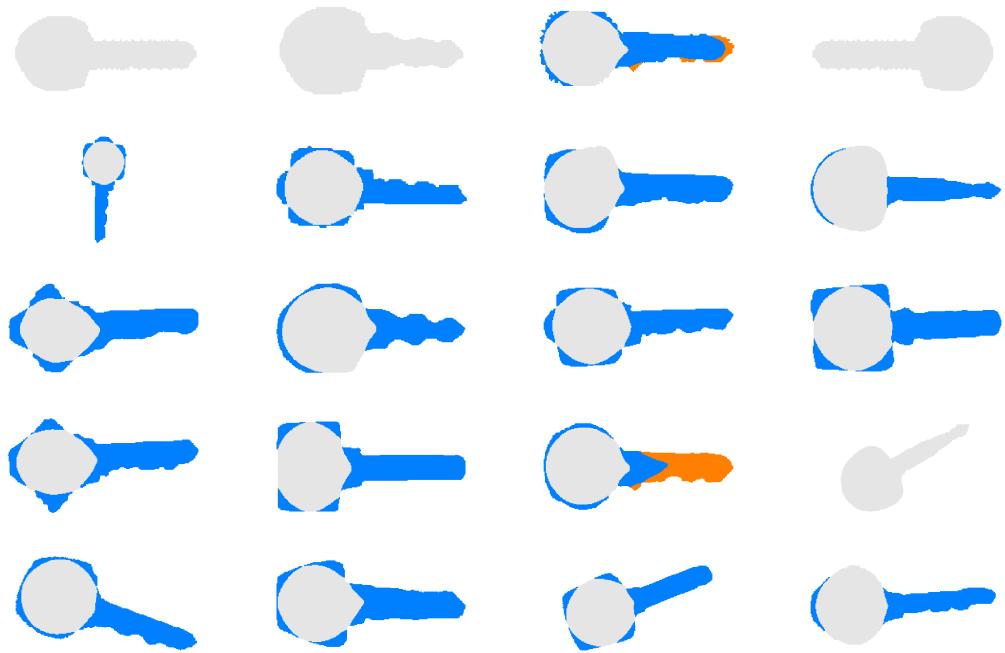


Figure B.30: Groupwise analysis results for category 29.

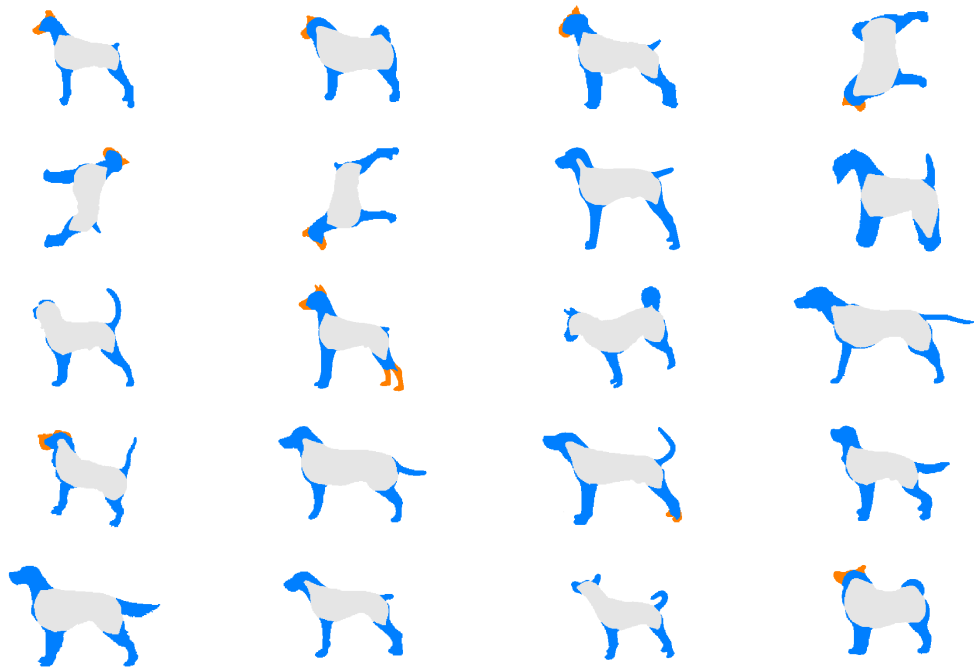


Figure B.31: Groupwise analysis results for category 30.

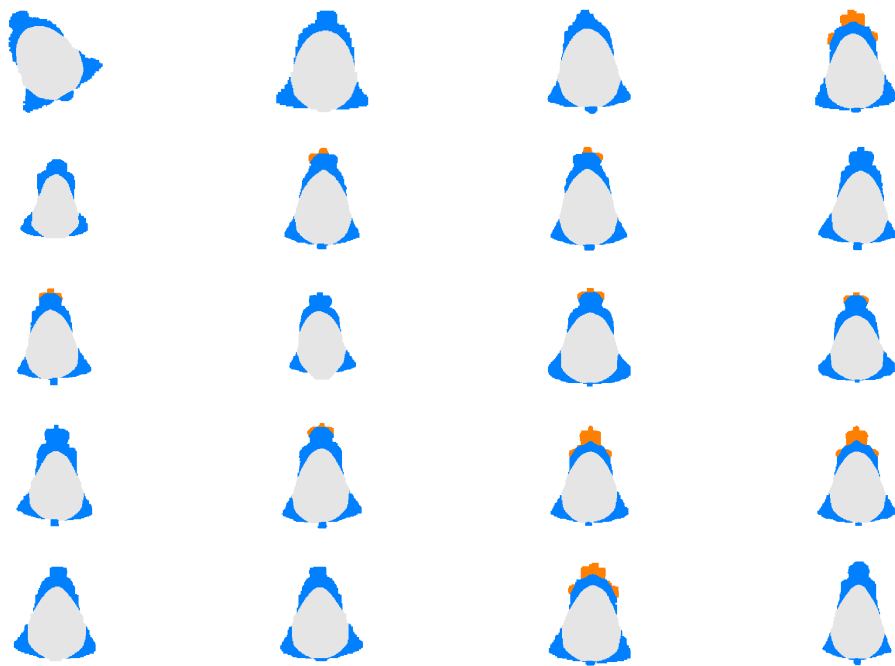


Figure B.32: Groupwise analysis results for category 31.

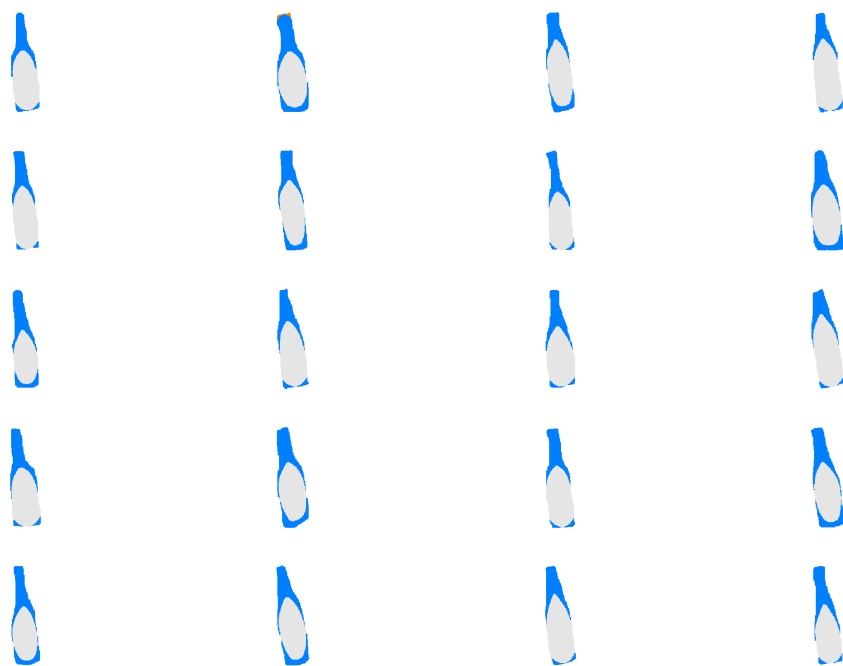


Figure B.33: Groupwise analysis results for category 32.

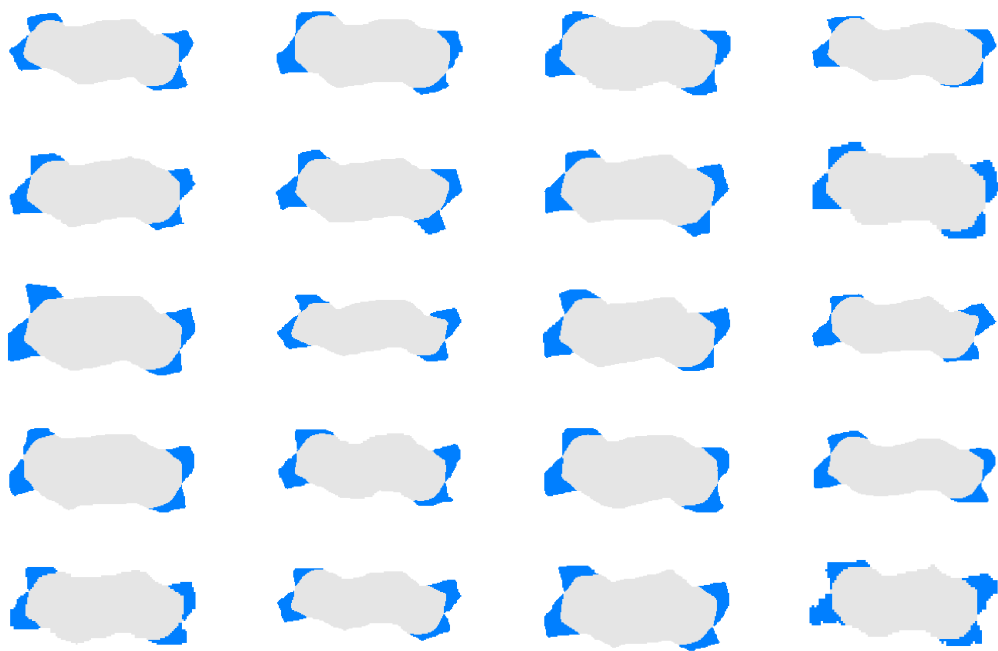


Figure B.34: Groupwise analysis results for category 33.

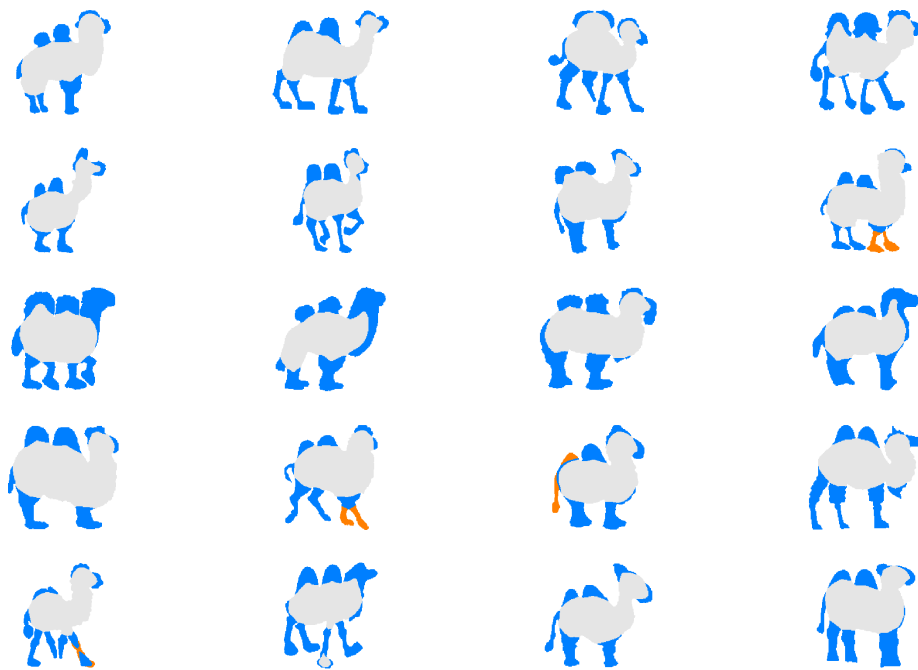


Figure B.35: Groupwise analysis results for category 34.



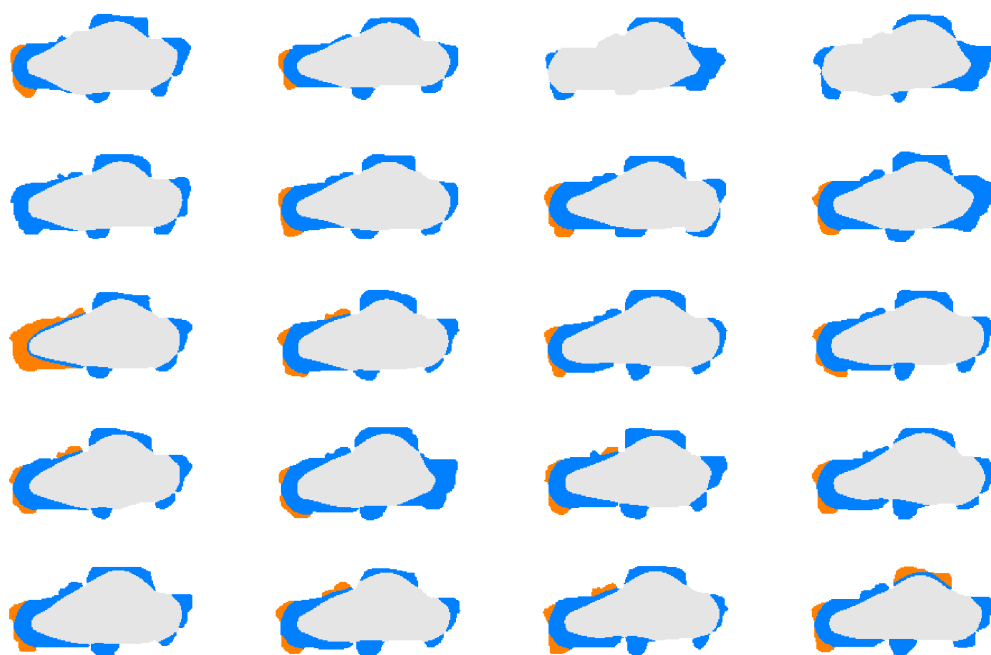


Figure B.36: Groupwise analysis results for category 35.

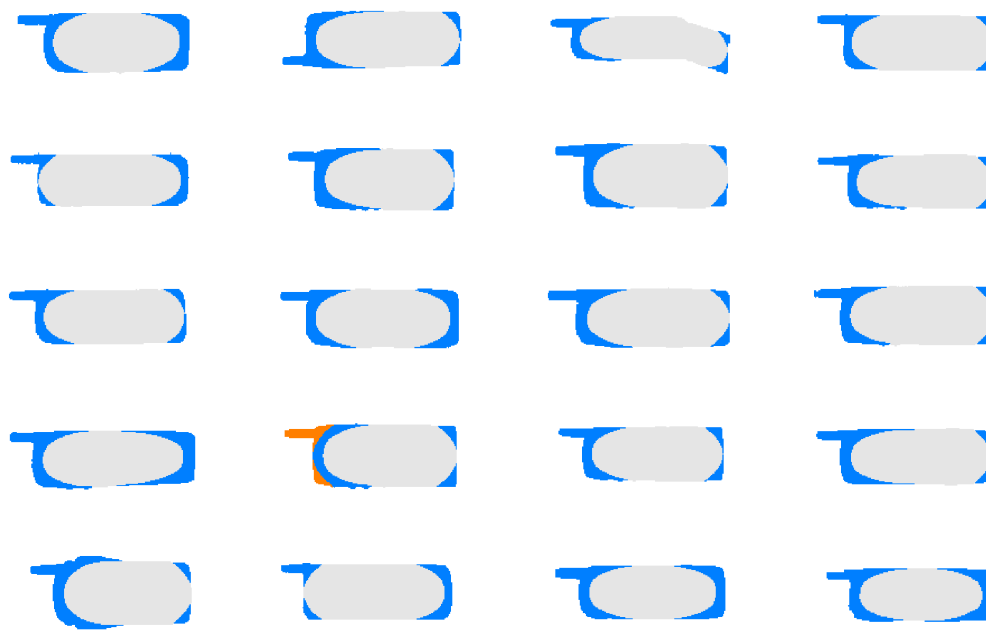


Figure B.37: Groupwise analysis results for category 36.

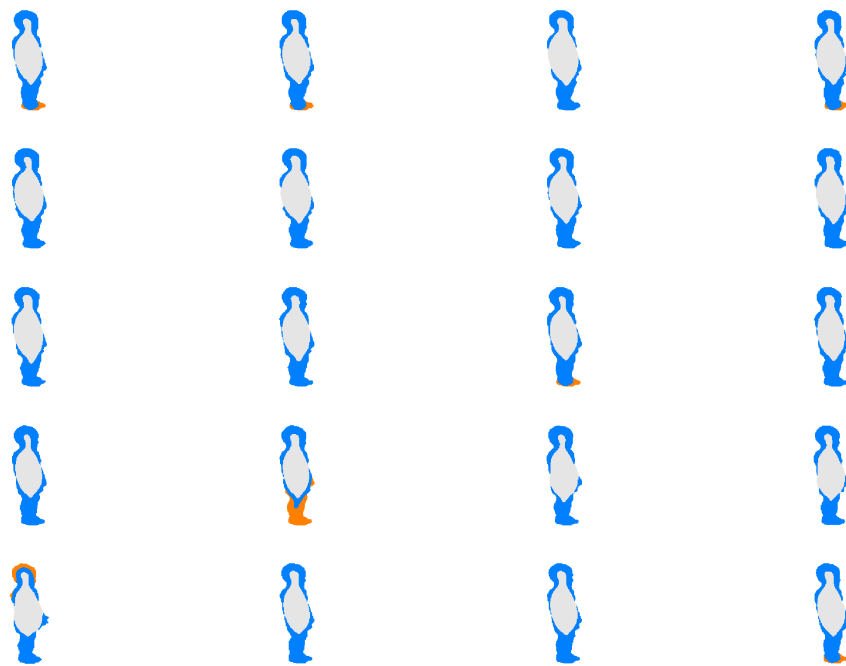


Figure B.38: Groupwise analysis results for category 37.



Figure B.39: Groupwise analysis results for category 38.

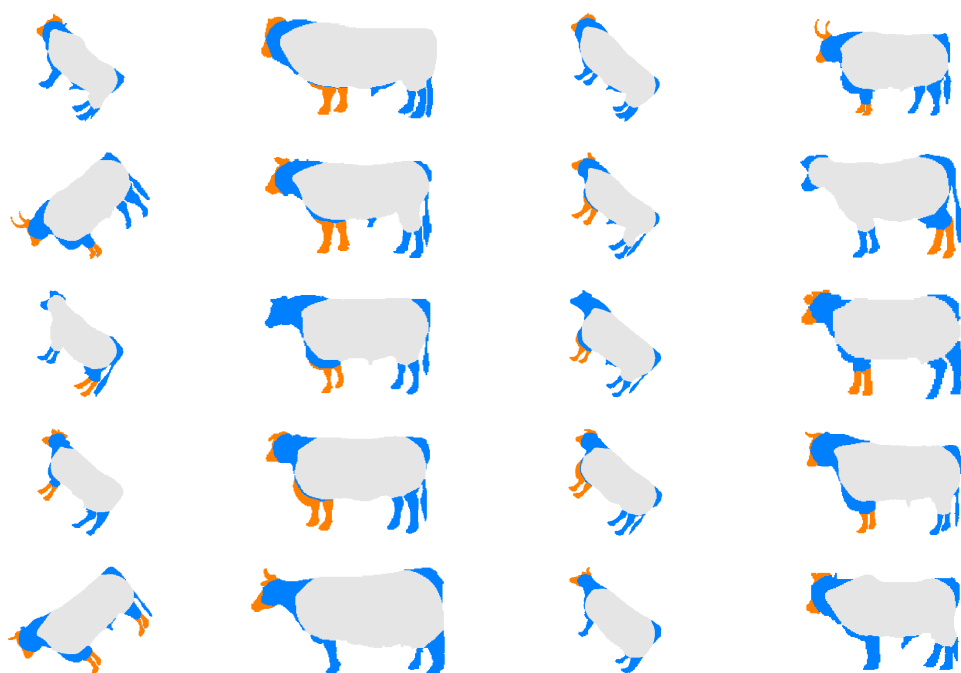


Figure B.40: Groupwise analysis results for category 39.

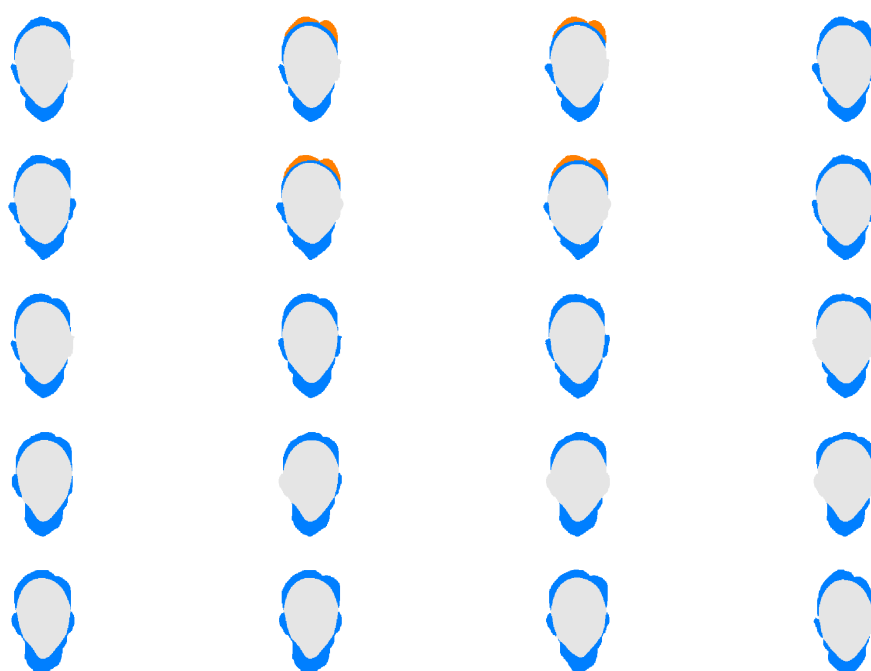


Figure B.41: Groupwise analysis results for category 40.

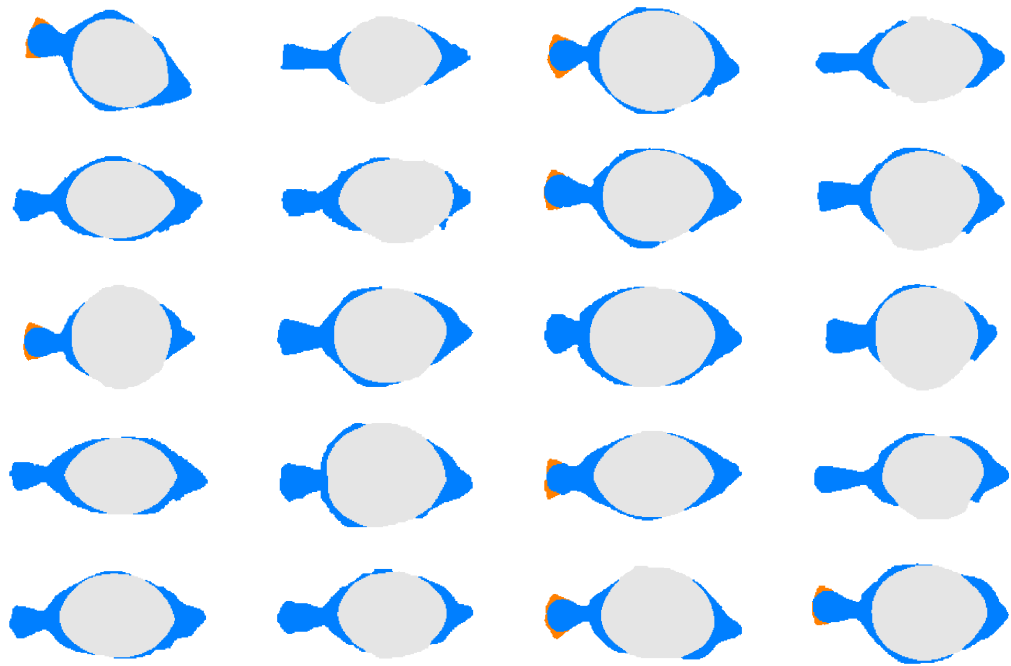


Figure B.42: Groupwise analysis results for category 41.

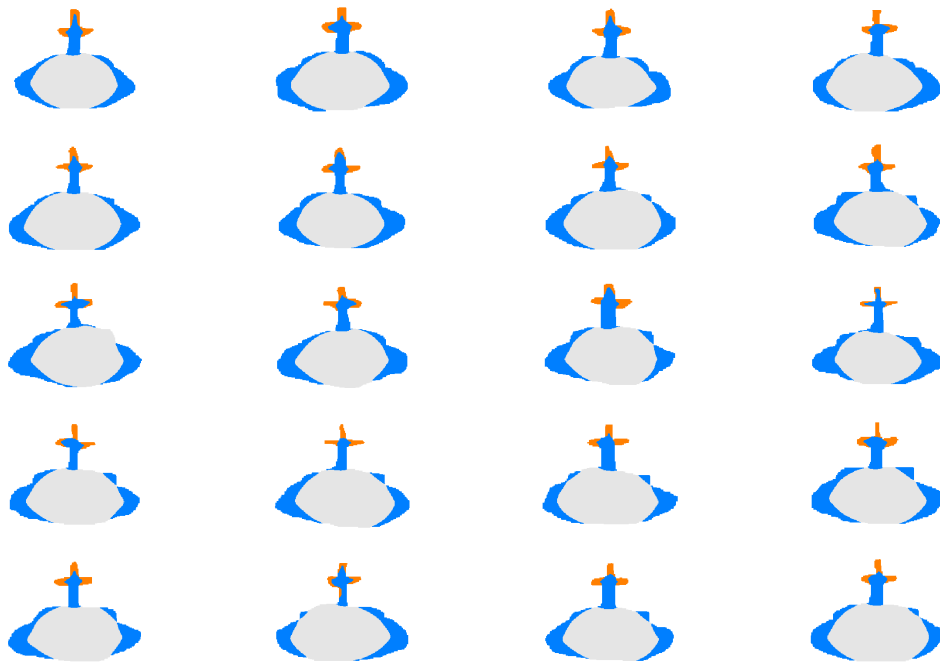


Figure B.43: Groupwise analysis results for category 42.



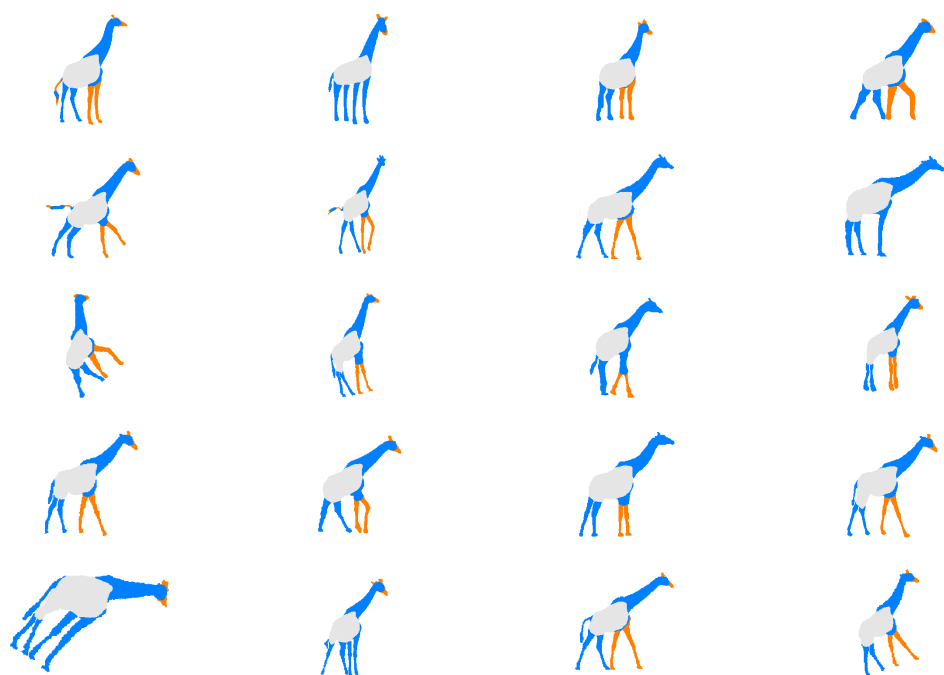


Figure B.44: Groupwise analysis results for category 43.

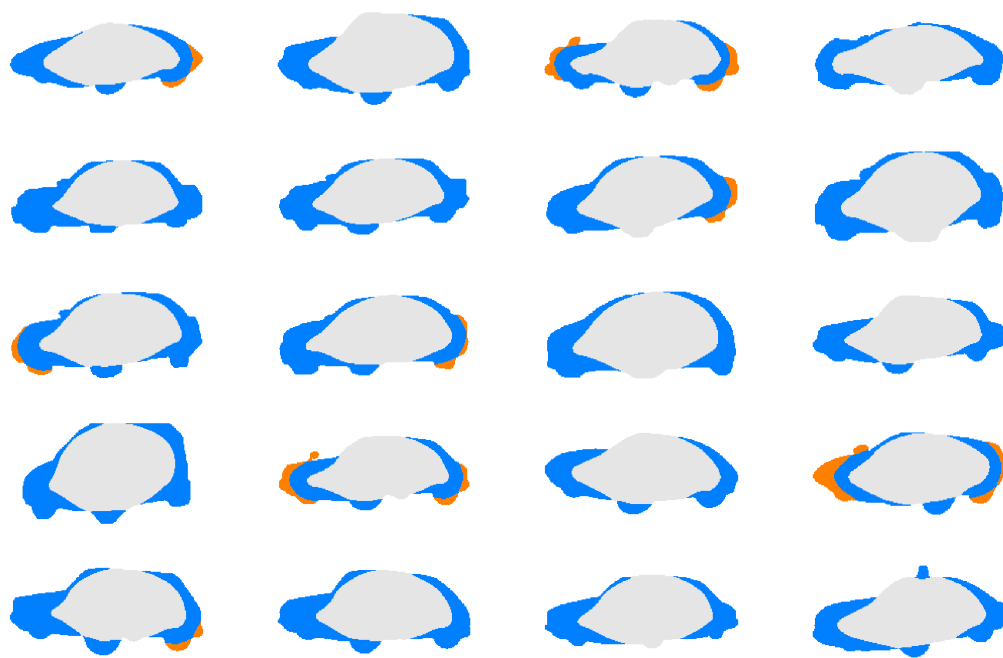


Figure B.45: Groupwise analysis results for category 44.

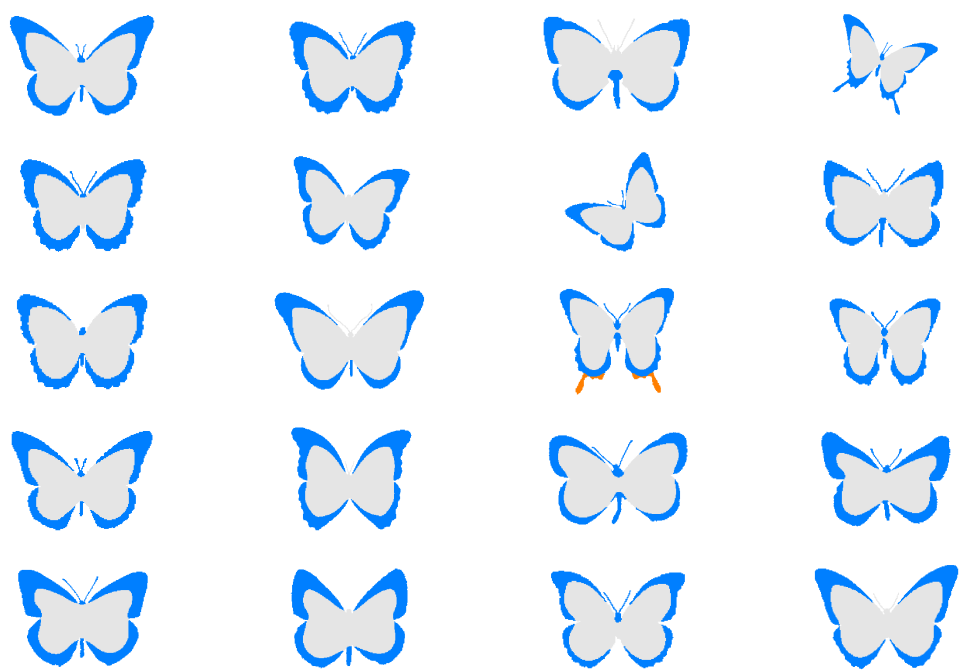


Figure B.46: Groupwise analysis results for category 45.

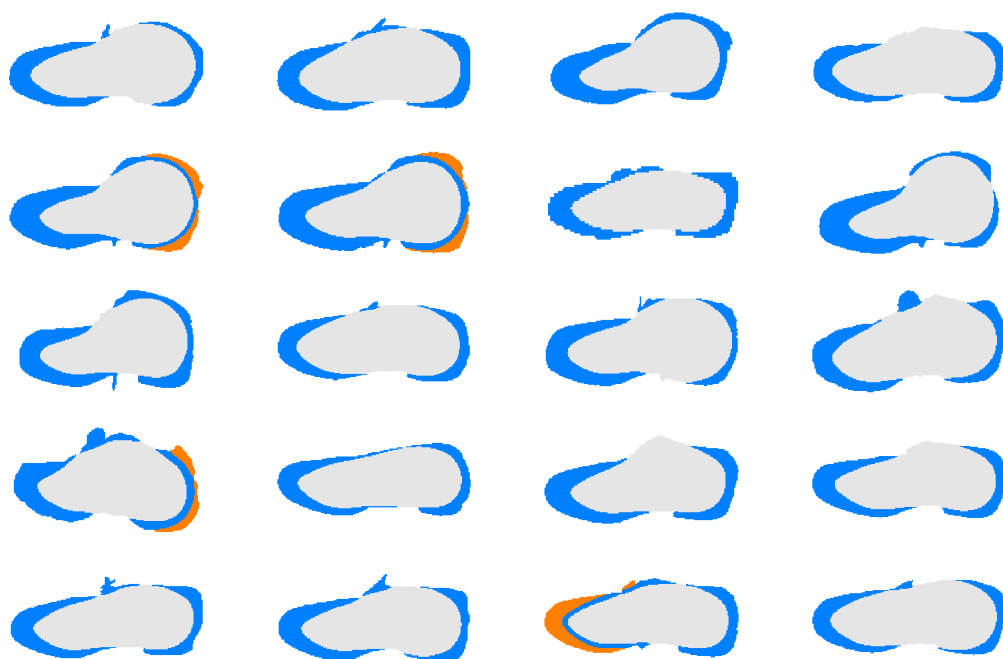


Figure B.47: Groupwise analysis results for category 46.



Figure B.48: Groupwise analysis results for category 47.



Figure B.49: Groupwise analysis results for category 48.



Figure B.50: Groupwise analysis results for category 49.

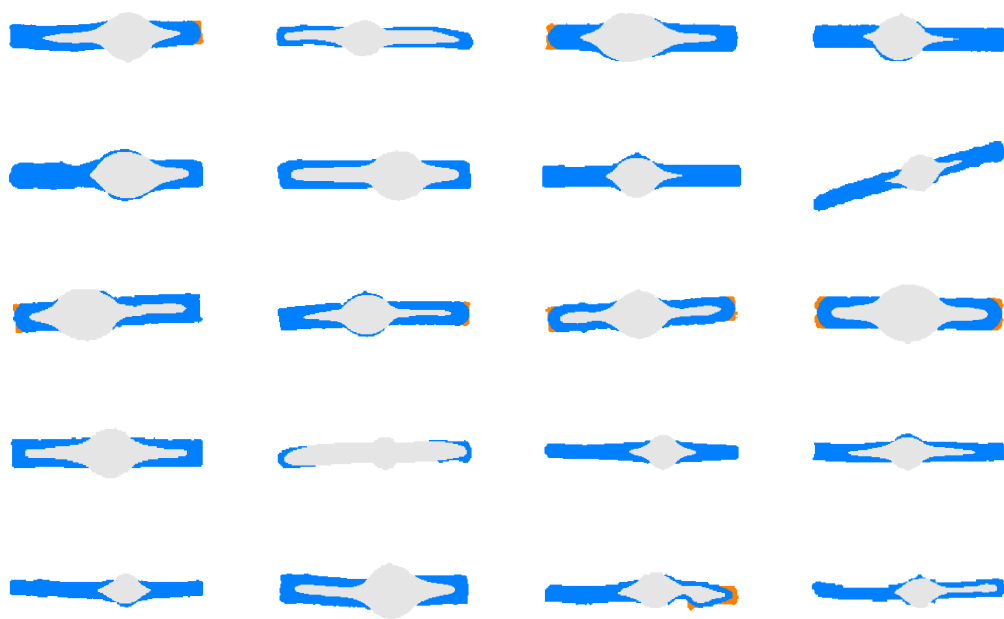


Figure B.51: Groupwise analysis results for category 50.



## CURRICULUM VITAE

### PERSONAL INFORMATION

**Surname, Name:** Gençtav, Murat

**Nationality:** Turkish (TC)

**Date and Place of Birth:** 01.09.1984, Bor/Niğde

**Marital Status:** Married

**Phone:** 0 544 6548797

### EDUCATION

Degree	Institution	Year of Graduation
M.S.	Middle East Technical University	2010
B.S.	Yildiz Technical University	2007
High School	İstanbul Lisesi	2003

### PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2017-..	ASELSAN A.S.	Expert Software Engineer
2013-2014	HAVELSAN A.S.	Expert Software Engineer
2008-2012	Middle East Technical University	Research&Teaching Assist.

### PUBLICATIONS

Genctav, M. and Tari, S. Multi-parameter Mumford-Shah Segmentation, Research in Shape Analysis: WiSH2, Sirince, Turkey, June 2016. Genctav, A., Leonard, K., Tari,

S. Hubert, E., Morin, G., El-Zehiry, N. and Chambers, E. (Eds.), Springer International Publishing, 2018, 133-141.

Genctav, M., Genctav, A. and Tari, S. NonLocal via Local–NonLinear via Linear: A New Part-coding Distance Field via Screened Poisson Equation, *Journal of Mathematical Imaging and Vision*, 2016, 55, 242-252.

Tari, S. and Genctav, M. From a Non-Local Ambrosio-Tortorelli Phase Field to a Randomized Part Hierarchy Tree, *Journal of Mathematical Imaging and Vision*, 2014, 49, 69-86.

Tari, S. and Genctav, M. From a Modified Ambrosio-Tortorelli to a Randomized Part Hierarchy Tree, *Scale Space and Variational Methods in Computer Vision (SSVM'11)*, Ein-Gedi, Israel, 2011.

Yildiz, H.K., Genctav, M., Usta, N., Amasyali, M.f. and Diri, B. A New Feature Extraction Method for Text Classification, *Signal Processing and Communications Applications*, IEEE 15th (SIU'07), Eskisehir, Turkey, 2007.

Genctav, M. and Manguoglu, M. Enhancing Parallelism in Monte Carlo Techniques for Solving Large Sparse Linear Systems, *Applied and Computational Mathematics*, Int. Conf. on (ICACM'12), Ankara, Turkey, 2012.